

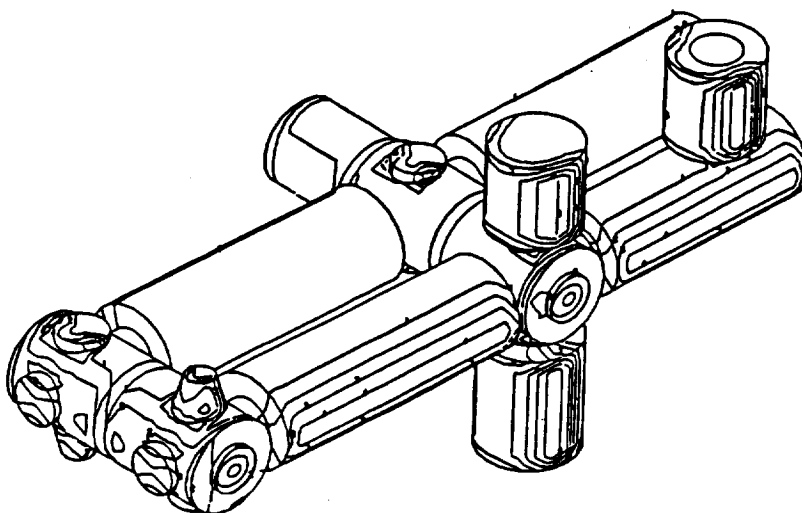
D180-30550-4

Users Guide
For Design Analysis Code BUMPER

SPACE STATION INTEGRATED WALL DESIGN AND PENETRATION DAMAGE CONTROL

by

A. R. Coronado, M. N. Gibbins, M. A. Wright, and P. H. Stern



Prepared for

National Aeronautics and Space Administration

July 1987

Contract NAS8-36426

Technical Management
NASA George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama
Structures and Dynamics Laboratory
Sherman L. Avans

Boeing Aerospace Company
Seattle, Washington

N88-10070

Unclas
0084733

33/18

(NASA-CR-179169) SPACE STATION INTEGRATED
WALL DESIGN AND PENETRATION DAMAGE CONTROL
(Boeing Aerospace Co.) 312 p Avail: NTIS
HC A14/MP A01 CSCL 22B

D180-30550-4

**Users Guide
For Design Analysis Code BUMPER**

SPACE STATION INTEGRATED WALL DESIGN AND PENETRATION DAMAGE CONTROL

by

A. R. Coronado, M. N. Gibbins, M. A. Wright, and P. H. Stern

Prepared for
National Aeronautics and Space Administration

July 1987

Contract NAS8-36426

**Technical Management
NASA George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama
Structures and Dynamics Laboratory
Sherman L. Avans**

**Boeing Aerospace Company
Seattle, Washington**

D180-30550-4
CONTENTS

	<u>Page</u>
1.0 INTRODUCTION	1
1.1 DOCUMENT ORGANIZATION/SECTION DESCRIPTION	1
2.0 BUILDING A MODEL	3
2.1 INPUT FILE FORMAT	3
2.2 MODELING CONVENTIONS	3
3.0 ANALYZING A MODEL	9
3.1 REFERENCE CONFIGURATION	9
3.2 GEOMETRY MODULE	9
3.3 RESPONSE MODULE	9
3.4 BUMPER MODULE	12
3.5 CONTOUR MODULE	12
4.0 PERFORMING SENSITIVITY STUDIES	13
4.1 MANUALLY CALCULATED FROM PREVIOUS RESULTS	13
4.2 STUDIES REQUIRING RERUNNING SOLUTION SEQUENCE	14
5.0 GENERATING CONTOUR PLOTS	17
5.1 THREAT CONTOUR PLOTS	17
5.2 DESIGN CONTOUR PLOTS	17
6.0 EXAMPLES	20
6.1 GEOMETRY	21
6.2 RESPONSE	21
6.3 BUMPER	29
6.4 CONTOUR	34
6.5 RPLOT	38
7.0 APPLICATION OF THEORY	41

7.1 HISTORY	41
7.2 DETERMINING THE PROBABILITY OF NO PENETRATION	41
7.3 GEOMETRY VERSION 2.51	41
7.3.1 Write Header	43
7.3.2 Calculate the Threat Data	43
7.3.3 Read in the Model File	46
7.3.4 Calculate the Normal Vectors	46
7.3.5 Calculate the Centroid Locations	46
7.3.6 Calculate the Surface Areas	46
7.3.7 Calculate the Radius of a Sphere Enclosing the Elements	46
7.3.8 Eliminate Back Side Elements	46
7.3.9 Transform the Nodal Coordinates and Centroid Locations	46
7.3.10 Sort on the Transformed X-Dimension	46
7.3.11 Sort on the Transformed Y-Dimension	48
7.3.12 Eliminate the Shadowed Elements	48
7.3.12 Output List of Exposed Elements and Associated Impact Angles	48
7.4 RESPONSE VERSION 2.0	48
7.4.1 Set Increments	51
7.4.2 Read in User Inputs	51
7.4.3 Set the Impact Angle	51
7.4.4 Set the Impact Velocity	51
7.4.5 Single Plate Penetration Function	51
7.4.6 Double Plate Penetration Function	51
7.4.7 PEN4	54

D180-30550-4	Page
7.4.8 Write Output	54
7.5 BUMPER VERSION 4.0	54
7.5.1 Read User Inputs	60
7.5.2 Read in the Geometry Data Base	60
7.5.3 Read in the Response Data Base	60
7.5.4 Set the Threat Data	60
7.5.5 Set the Impact Conditions for Element Nel	60
7.5.6 Calculate the Critical Diameter	61
7.5.7 Calculate the Flux of the Critical Diameter	61
7.5.8 $HISTORY(NEL) = HISTORY(NEL) + F(D_c(v_i, \beta_i)) * \cos(\beta_i) * \rho(\alpha_i, \theta_i)$	61
7.5.9 $ARHIS(NEL) = ARHIS(NEL) + \cos(\beta_i) * \rho(\alpha_i, \theta_i)$	61
7.5.10 $HISTORY(NEL) = HISTORY(NEL) * Area(NEL),$ $ARHIS(NEL) = ARHIS(NEL) * Area(NEL)$	61
7.5.11 Sum up the History and Arhis Arrays by Ranges	61
7.5.12 Calculate PNP for Each Range	61
7.5.13 Write PNP and Effective Area Out	62
7.5.14 SUPERTAB Output File	62
7.6 CONTOUR VERSION 2.0	62
7.6.1 Read in User inputs	62
7.6.2 Set Vessel Wall Thickness	62
7.6.3 Set Shield Thickness	62
7.6.4 Ratio OK	62
7.6.5 Build Response Array	64
7.6.6 Inside Range	64
7.6.7 Sum History by Specific Range and PID	64

7.6.8 Calculate the PNP for the Specific Range and PID	64
7.6.9 Write Out Ts, S, Tb, PNP	64
REFERENCES	65
APPENDIX A - SUPERTAB Universal File Format	67
APPENDIX B - Source Code for GEOMETRY Module	B-1
APPENDIX C - Source Code for RESPONSE Module	C-1
APPENDIX D - Source Code for BUMPER Module	D-1
APPENDIX E - Source Code for CONTOUR Module	E-1

(D180-30550-4

FIGURES

		<u>Page</u>
1.0-1	Analysis Flow	2
2.2-1	Modeling Conventions	4
2.2-2	Dynamic Receiver Model	6
3.1-1	Space Station Geometry Model, 5000 Elements	10
3.3-1	Correlations of Predictions Versus Test Data	11
4.1-1	Results of Orbital Time Sensitivity Study	15
4.2-1	Results of Orientation Sensitivity Study	16
5.1-1	Threat Contour Plot - Debris Environment	18
5.1-2	SUPERTAB Postprocessing Input for Threat Contour Plot	19
5.2-1	Design Contour Plot - Debris Environment	20
6.1-1	Example of GEOMETRY Run - Debris	22
6.1-2	Debris Threat Generation	23
6.1-3	Example of GEOMETRY Run - Meteoroids	24
6.1-4	Meteoroid Threat Generation	25
6.2-1	Example of RESPONSE Run	26
6.2-2	RESPONSE Summary File	28
6.3-1	Example of BUMPER Run	30
6.3-2	BUMPER Summary File	32
6.3-3	Example of Orbital Altitude Calculation	33
6.4-1	Example of CONTOUR Run	35
6.4-2	Sample CONTOUR Summary File	37
6.5-1	Example of RPLOT Run	39
6.5-2	Sample RPLOT Output File	40

D180-30550-4

		<u>Page</u>
7.3-1	GEOMETRY Flow Chart (Version 2.0)	44
7.3-2	Transformation of Velocity Distribution to Approach Angle Distribution	45
7.3-3	Transformation of Nodes and Sorting of Elements	47
7.3-4	Geometry Data Base File Structure	49
7.4-1	RESPONSE Flow Chart (Main Program.)	50
7.4-2	Example of MAT.PRP File (Material Property File.)	52
7.4-3	Comparison of Penetration Resistance to Meteoroids of Single Wall and Double Wall Structures	53
7.4-4	Penetration Function With Burch Equations	55
7.4-5	Regression Penetration Function	56
7.4-6	Penetration Function With PEN4	57
7.4-7	Response Data Base File Structure	58
7.5-1	BUMPER Flow Chart	59
7.6-1	CONTOUR Flow Chart	63

D180-30550-4

GLOSSARY

BAC	Boeing Aerospace Company
BUMPER	The family of computer codes developed under this contract or the specific module for calculating the probability of no penetration.
Critical Diameter	The diameter of a particle that just penetrates a configuration for a given impact velocity and angle.
EID	element identification number
FEM	finite element model
Flux	The number of particles passing through unit area per unit time.
IELM	Parameter controlling maximum number of elements in common block.
MLI	multilayer insulation
MSFC	Marshall Space Flight Center
NASA	National Aeronautics and Space Administration
NASTRAN	NASA structural analysis finite element code
NASUP	NASTRAN to universal file format translator
NEL	Nth element
PC	personal computer
PID	element's property identification number
PNP	probability of no penetration
RPLOT	translates RESPONSE binary output file to a formatted file suitable for plotting.
SUPERTAB	Graphic preprocessor and postprocessor for finite element analysis by Structural Dynamics Research Corporation of San Diego, California.
SSCE	Space Station critical element

D180-30550-4

Universal File Format	Machine-independent format for transmitting data to and from SUPERTAB (see app. A)
VAX	Mainframe computer build by Digital Equipment Corporation.
VMS	Virtual Memory System, operating system for VAX computers.

1.0 INTRODUCTION

The analysis code BUMPER executes a numerical solution to the problem of calculating the probability of no penetration (PNP) of a spacecraft subject to man-made orbital debris or meteoroids impact. The term "BUMPER" as used in this document, refers to both the overall family of codes as well as the specific computer program BUMPER. This document provides step-by-step procedures and theory for performing such an analysis. The examples provided in the text reflect our approach to analyzing Space Station structure, but the principles can be applied to any structure that can be modeled with finite elements. Advantages of this approach include (1) impact shielding of one element by another (shadowing) is considered, (2) users can specify various shield configurations over the spacecraft exterior to reflect design requirements, and (3) the effects of changing spacecraft flight orientation and orbital altitude can be determined.

The codes were developed on a DEC VAX 11/780 computer that uses the VMS operating system. They are written in Fortran 77 with no VAX extensions.

To help illustrate the steps involved, we carry a single sample analysis through the users guide. The example is our Space Station reference configuration used throughout contract performance. The finite element model (FEM) of this configuration is relatively complex but demonstrates many BUMPER features.

A flow chart for a complete analysis is shown in figure 1.0-1. The path indicated by solid lines is required to perform a fundamental analysis resulting in an overall PNP for the spacecraft. The dashed lines show how ancillary sensitivity studies on design variables are performed.

1.1 DOCUMENT ORGANIZATION/SECTION DESCRIPTIONS

Section 2.0 describes computer tools and guidelines for constructing a FEM for the space structure under consideration.

Section 3.0 carries through the steps in a fundamental analysis of the model constructed in section 2.0.

Section 4.0 describes the methods used to analyze the sensitivity of PNP to variations in design.

Section 5.0 suggests ways for developing contour plots of the sensitivity study data.

Section 6.0 provides additional BUMPER analysis examples and includes FEMs, command inputs, and data outputs.

Section 7.0 describes the mathematical theory used as the basis for the code and illustrates the data flow within the analysis.

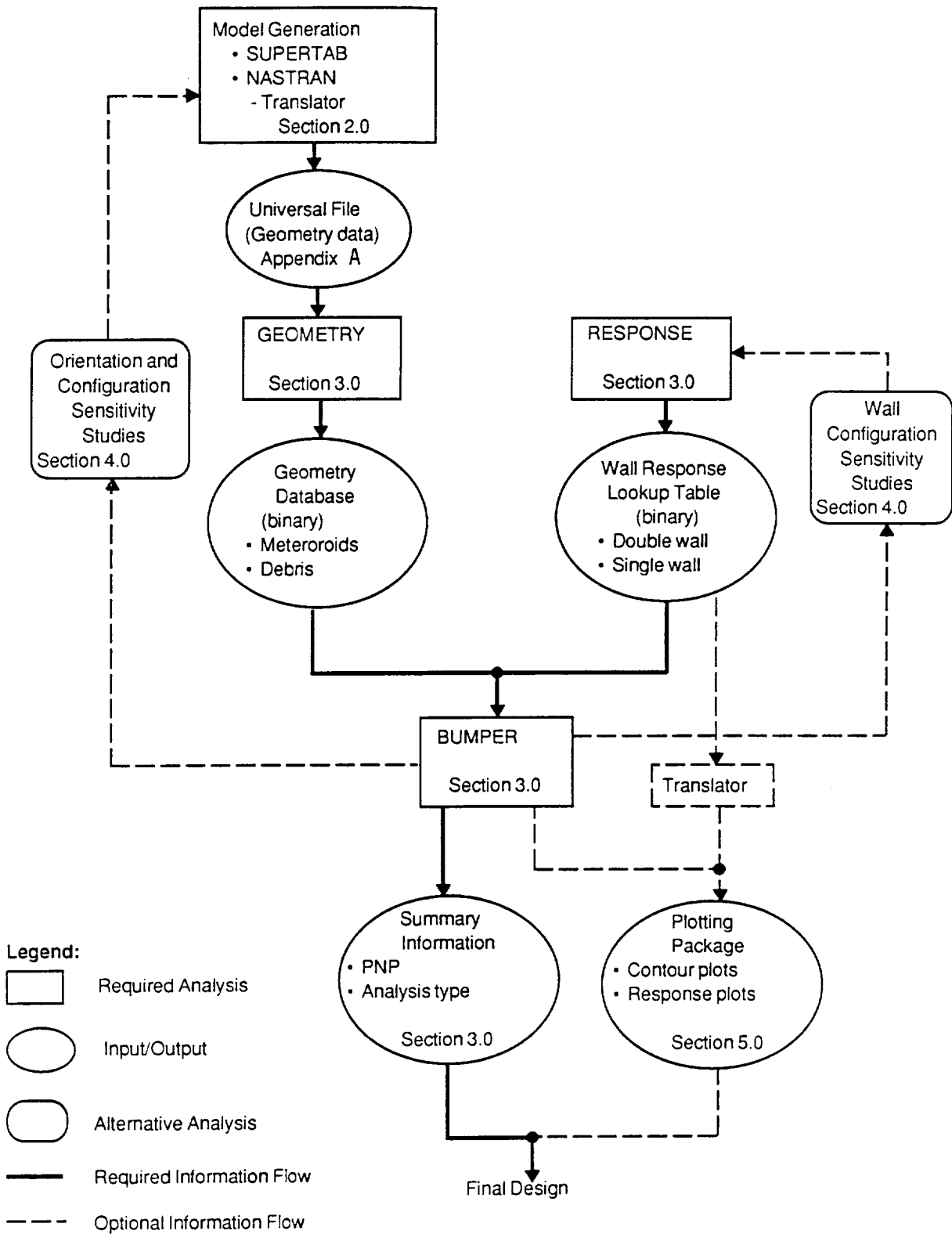


Figure 1.0-1. Analysis Flow

870512-1

2.0 BUILDING A MODEL

The geometry and orientation of the spacecraft is defined mathematically through the use of a FEM. The conventions for building such a model are described below.

2.1 INPUT FILE FORMAT

The FEM model used to represent the spacecraft geometry and configuration may be built using conventional FEM generation techniques. Under this contract, the FEM preprocessor and postprocessor SUPERTAB was used for building and debugging the model and later for analysis results presentation. This was done for convenience; the model generation code or method used is not critical. Other FEM codes such as ANSYS or SPAR may be used to build the model but will require a translator to convert the node and element connectivity information into the proper format. After generation the completed model is written to an analysis input file in the SUPERTAB Universal File Format. This is the format the analysis code GEOMETRY uses. For convenience a NASTRAN to Universal File Format translator named NASUP is available from NASA/MSFC. A model built using another format will require a translator to convert it to the Universal File Format shown in appendix A of this document.

2.2 MODELING CONVENTIONS

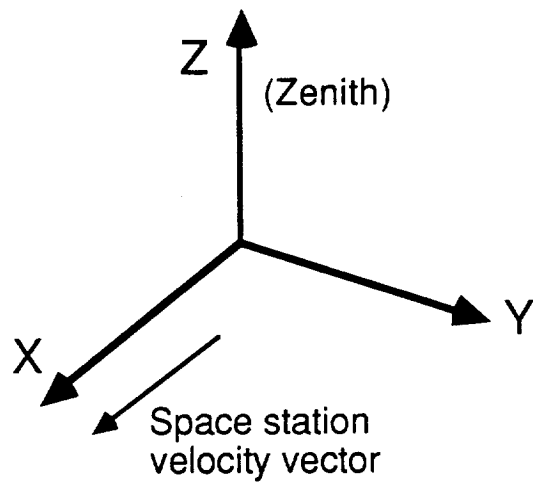
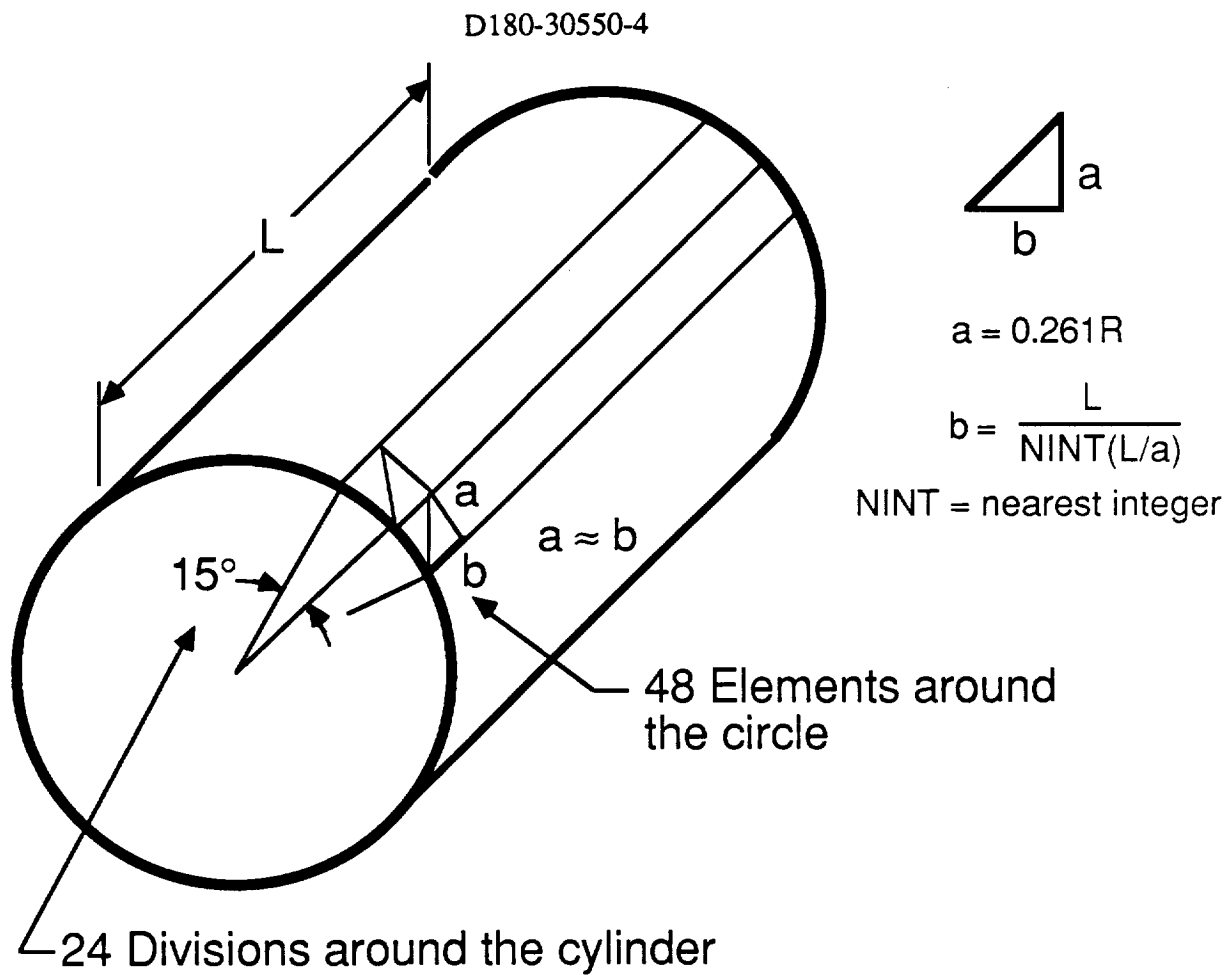
Nodes must all be defined in one global Cartesian coordinate system, as shown in figure 2.2-1. GEOMETRY does not check to see if more than one system is used. The positive X-axis must be parallel to the velocity vector, and the positive Z-axis must point toward the zenith (away from Earth). Failure to follow these conventions will invalidate the model and the analysis.

Only triangular elements may be use to build the FEM. The most reliable results are obtained using equilateral triangles, but right isosceles triangles may be more convenient when modeling cylindrical components. In any case, the element's aspect ratio should be as close to (1:1) as practical. Long, thin triangular elements may lead to erroneous results. This is because the hidden surface algorithm determines if an element's centroid is hidden by another element when deciding whether or not an element is exposed.

When modeling cylindrical sections, at least 24 divisions (48 elements) should be used around the circumference, as shown in figure 2.2-1. Axial divisions then should be chosen to optimize the aspect ratio (1:1). The user should be aware that sensitivity studies of element sizes were done for horizontal cylindrical sections only. Also, the user should ensure an adequate model by verifying that the calculated PNP does not change significantly with changes in mesh size.

All dimensions must be specified in meters. Using any other dimensional system will require code revisions.

The node numbering order defines, by the right-hand rule, the



Model Coordinate System

Figure 2.2-1. Modeling Conventions.

direction of the element's outward normal vector. The code uses the nodes in order 1, 2, 3. This vector is used to determine whether or not the element faces toward the threat, and it is extremely important to an accurate analysis. Ordinarily the outer surface of the FEM consists of elements all having outward normal vectors, but in special cases where impacts on internal surfaces (particles passing through openings) are important, additional elements with inward pointing normal vectors may be required. Figure 2.2-2 shows a Solar Dynamic Receiver model in which both outward and inward normal elements are used. Debris and meteoroids could strike the receiver's outer surface or pass through the orifice and strike the more sensitive inner surface.

A maximum of 10 property identifications (PID), numbered 1 through 10 sequentially, may be defined. The code can be easily modified to allow for more PIDs. Each PID defines one unique wall, shield, spacing, and multilayer insulation (MLI) configuration. Therefore, each model may contain only 10 different wall configurations. These configurations may be either a conventional double plate (shield and pressure wall) arrangement, with or without 30 layers of MLI or a single thick wall. The code assumes all elements are parallel flat plates without curvature for penetration analysis.

The maximum number of elements allowed depends on the computer system. For a mainframe, the practical maximum is approximately 8000 elements; for an IBM-compatible personal computer (PC), the practical maximum is approximately 400 elements. The large data files required when analyzing a small model for meteoroid impacts may preclude using this code on a PC due to memory limitations.

The PNP will be calculated for specific ranges of element ids. Therefore several items should be kept in mind while building the model and determining the element numbering scheme. All elements representing a single Space Station critical element (SSCE) should be numbered within a single range of numbers and sequentially if possible, for example, number all the elements representing common module 1 in the range 1000-1999 and for node 3 in the range 5000-5999. This allows determination of PNP for each SSCE. Within this range of element IDs any combination of PIDs, up to a maximum of 10, may be used to accurately described differences in wall construction.

Many customary FEM modeling limitations do not apply here. Coincident, extra, and unattached nodes will not cause premature code failure. The model does not require constraint against translations or rotations, as it is used only to define spacecraft geometry. Structures that are in reality attached need only be modeled as adjacent to each other. An important limitation is that, if a structure normally shadows (hides) another's view of a threat, it must be modeled with sufficient accuracy and resolution to represent that shadowing. Any elements totally enclosed and therefore completely shielded from all threats by other elements will not affect the solution.

The limits on elements may be changed, if necessary, to allow

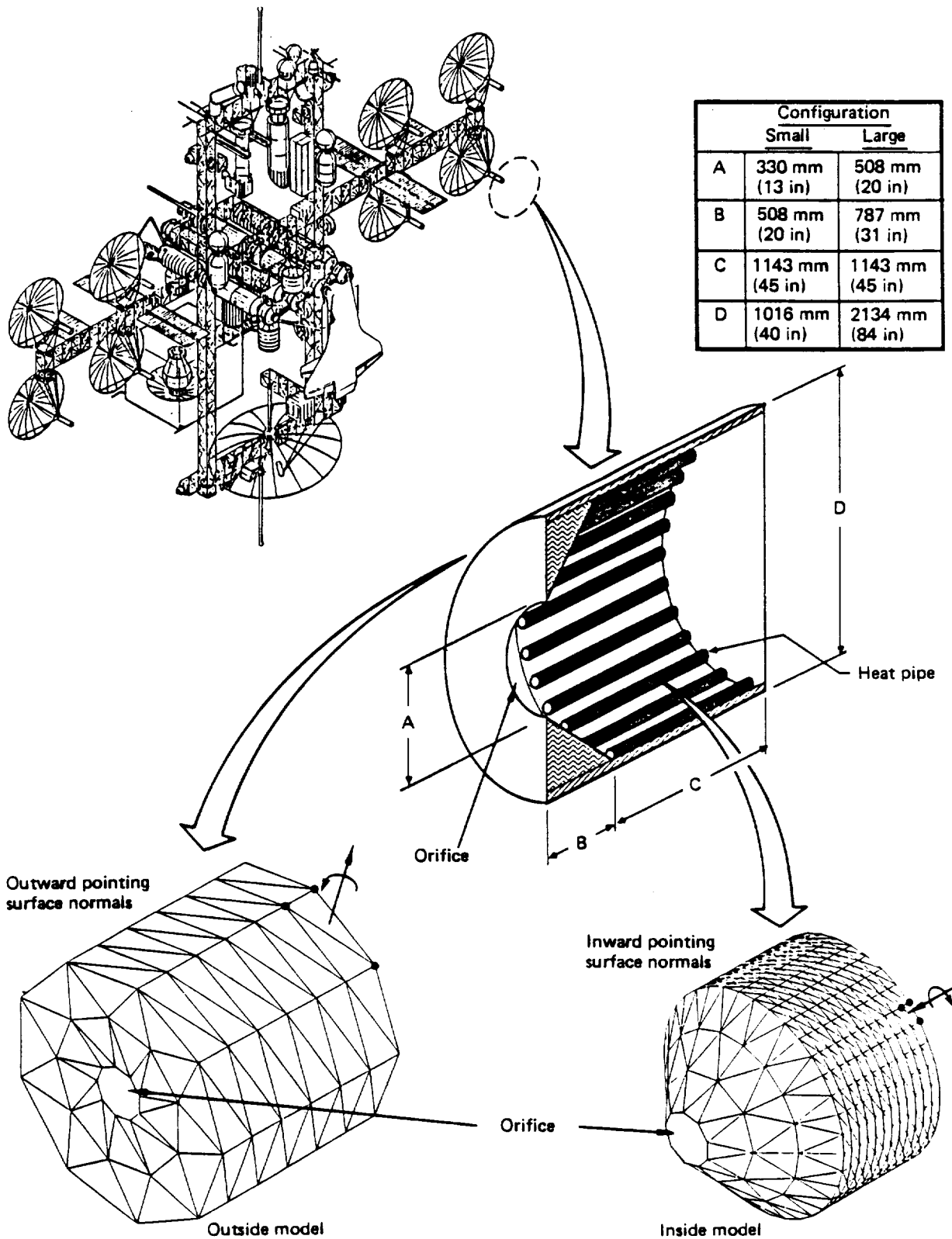


Figure 2.2-2. Dynamic Receiver Model

analysis of larger models. This may be done by modifying the parameter IELM in the files COMMON1.BLK, and COMMON2.BLK and recompiling the GEOMETRY and BUMPER codes. A careful user should perform test cases to ensure numerical limitations or instabilities are not encountered. This can be done by rerunning the analysis for models previously run under the unmodified software to ensure no differences in the outputs occur.

To change the number of PIDs both the source code of RESPONSE and the COMMON2.BLK file must be modified. In the RESPONSE source code the dimension statement for RTABLE must be changed to the new values and the code recompiled. In the COMMON2.BLK file the dimensions of the RESPONSE array should match those of RTABLE. The BUMPER code then must be recompiled.

This page left intentionally blank.

3.0 ANALYZING A MODEL

3.1 REFERENCE CONFIGURATION

To demonstrate the features of BUMPER and to illustrate design and analysis methodology, we selected the model of the reference configuration shown in figure 3.1-1; this model is described more fully in the Final Report (ref. 1). The reference configuration is representative of a typical Space Station configuration but does not represent any specific design.

Figure 1.0-1 shows the steps necessary to analyze the model described in section 2.0. Section 6.0 contains an example of the questions and menu options asked by each analysis module, what they represent, and typical responses. The required solution sequence is described in the following sections in the normal order of execution.

3.2 GEOMETRY MODULE

After the model has been built and verified, the geometry data must be written to a file using the SUPERTAB Universal File Format. GEOMETRY then reads in this information and creates the geometry data base for either meteoroid or debris analysis. Two geometry data bases are required for a complete analysis, one for debris and one for meteoroids. The data bases need only be recreated if the spacecraft geometry or orientation with respect to the velocity vector changes. This is the most computer-intensive portion of the analysis. If possible, it should be performed using a batch mode to minimize cost.

3.3 RESPONSE MODULE

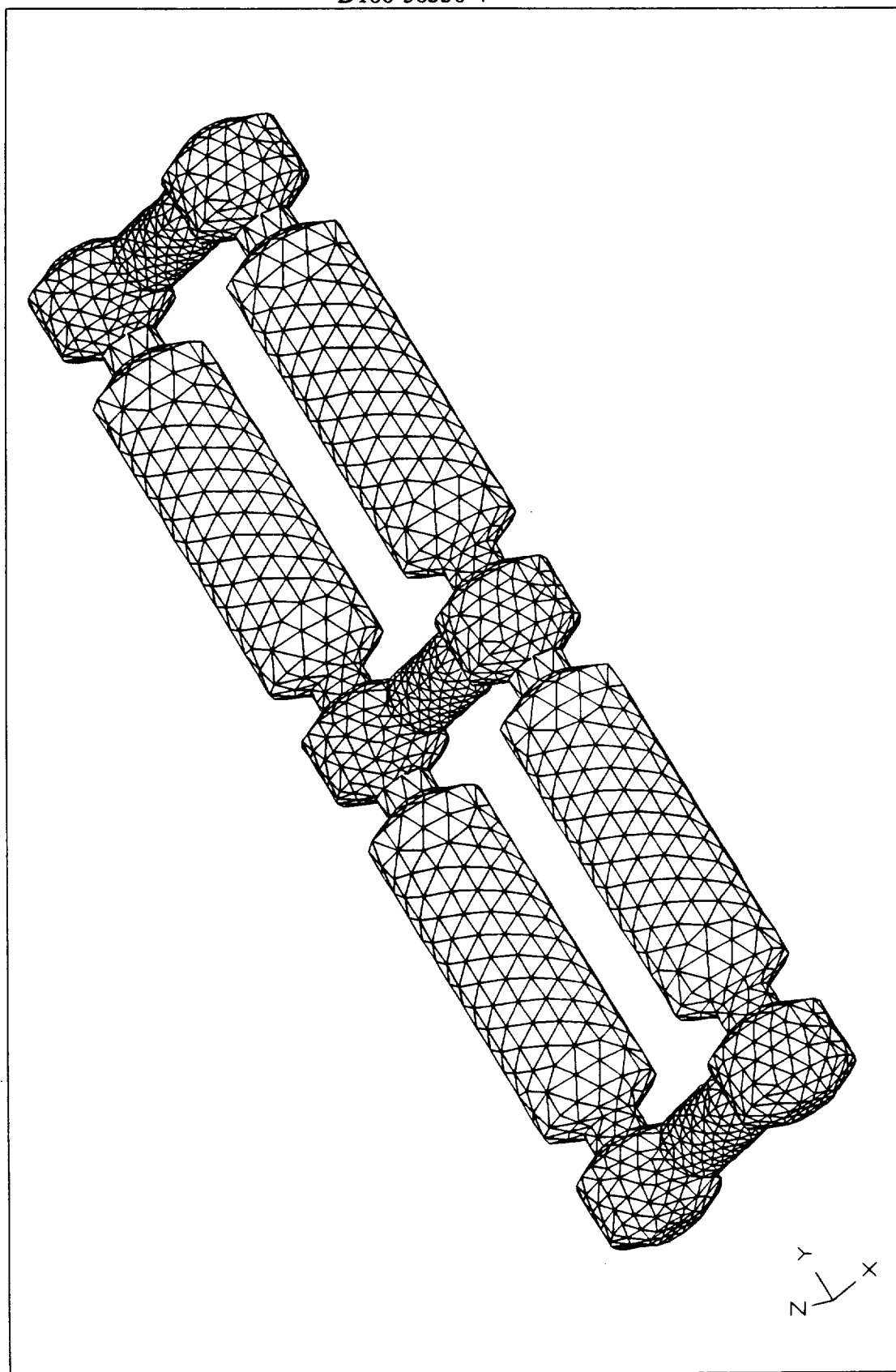
After the geometry data bases have been produced, the wall penetration response data bases are created. Two data bases are required for a complete analysis, one for debris and one for meteoroids. RESPONSE creates these data bases for up to 10 unique wall configurations. These configurations are identified through the use of element property identifications (PID) in the FEM. RESPONSE does not have available any information about the model; therefore, the user must keep track of the number of PIDs and their relationship to the actual spacecraft. RESPONSE will prompt the user with various questions to determine the wall configurations for a particular analysis type. The results of the analysis are written to a user-defined file in binary format. The file contains the critical diameter for each wall configuration as a function of impact velocity and impact angle. A computer program named RPLOT is available from MSFC that converts the binary information to a form suitable for plotting. This is useful for illustrating the correlation between predicted wall response versus test data, as shown in figure 3.3-1. As a general rule, the user should examine the response plots for each wall configuration to ensure that the results are reasonable. RESPONSE is relatively inexpensive to use and therefore normally is run in an interactive mode. Examples of typical questions and inputs to RESPONSE and RPLOT are given in Section 6.0.

PRECEDING PAGE BLANK NOT FILMED

12-DEC-85

SDRC--I-DEAS 2.5B: Output Display

SPACE STATION INTEGRATED WALL

*Figure 3.1-1. Space Station Geometry Model, 5000 Elements*

ADP SM-1 TEST RESULTS

NORMAL IMPACTS

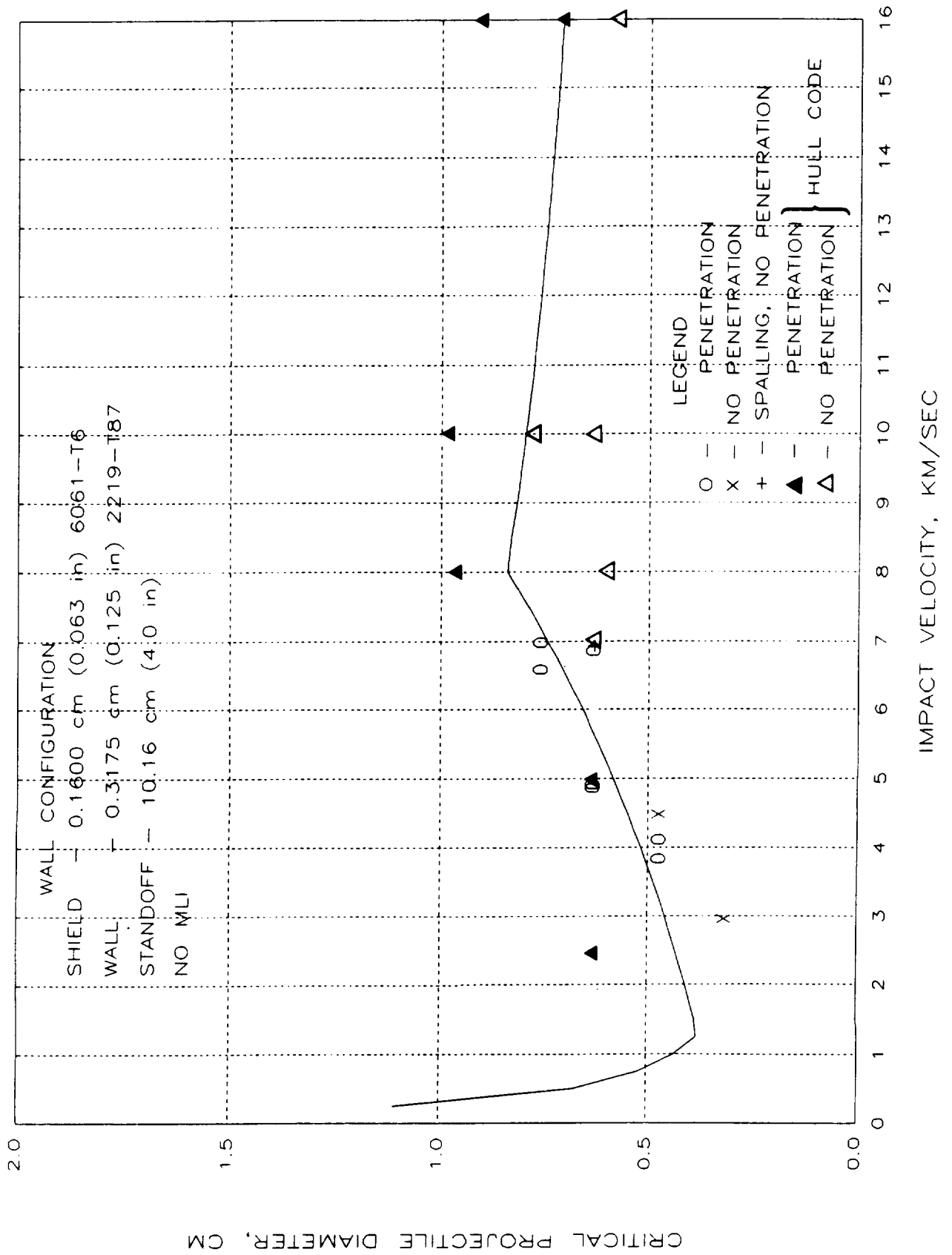


Figure 3.3-1: Correlations of Predictions Versus Test Data

3.4 BUMPER MODULE

Once the geometry and response data bases have been created, the probability of no penetration (PNP) is calculated using the BUMPER module. BUMPER will have to be run twice to perform a complete analysis, once for debris and once for meteoroids. The results from the two analyses are multiplied together to determine the overall PNP for both debris and meteoroids. The code prompts the user for information concerning the analysis type, summary filename, exposure time, operating altitude, geometry and response data base filenames, and the element ID ranges. The element ID ranges define the individual components (SSCE) for PNP calculation. The code also determines the PNP for the overall model. The results are then written both to the screen and to the user-defined summary file. Additionally BUMPER determines the effective area of the components and overall model. This information can be used in a simplified analysis to determine the number of particles of diameter D or greater per unit time affecting a specific component.

Additionally, BUMPER can produce the probability of penetration values per surface area for each element, which SUPERTAB can then plot as contours on the geometry model. These contours show the relative vulnerability of various Space Station external areas. This procedure is discussed in more detail in section 5.0

3.5 CONTOUR MODULE

BUMPER calculates the PNP for each component for a specific wall configuration. Sometimes, however, it is required to know how changes in wall configuration affect the PNP of an individual component. CONTOUR produces a data base that may be used with user-supplied software to produce design contour plots. These plots show the relationship between shield and vessel wall thickness and PNP for a fixed shield standoff.

The code is similar to BUMPER, with the exception being that the RESPONSE code has been incorporated as a subroutine. The code uses the same geometry data base as BUMPER. CONTOUR, however, is limited to the case of one PID and one element ID range. This topic is discussed further in section 5.0.

4.0 PERFORMING SENSITIVITY STUDIES

Two types of sensitivity studies are typically required, (1) those that can be calculated manually from previous results and (2) those that require rerunning part of the solution sequence. Both will be covered in this section.

4.1 MANUALLY CALCULATED FROM PREVIOUS RESULTS

Studies falling into this category measure the effects on PNP of simple variations in orbital time, surface area, or flux. To begin, a PNP value must be calculated using the solution sequence shown in figure 1.0-1 and described in section 3.0. The fundamental equation for calculating PNP is

$$\text{PNP} = \text{EXP} [-\lambda * T] \quad (1)$$

where λ is the average number of penetrating particles per unit time and T is the exposure time. λ is defined by equation 2

$$\lambda = A_j * \cos(B_i) * F(D_c(V_i, B_i)) * P(\theta_i, \alpha_i) * I_s \quad (2)$$

where A_j is the element's surface area, $F(D_c(V_i, B_i))$ is the flux of the critical projectile D_c with impact velocity V_i and impact angle B_i , $P(\theta_i, \alpha_i)$ is the probability of the threat described by polar angles θ_i and α_i occurring and I_s is an indicator equal to 1 if the element is exposed and 0 if not. NT is the total number of threat angle cases and NELM is the number of elements in the range of interest (SSCE).

Scaler changes to the A , T , and F terms allow the PNP to be recalculated manually. Scaler changes to A imply that all the elements area change by a constant amount. Scaler changes to F imply that the flux of all particles changes by a constant amount. Scaler changes to T imply a simple change in the exposure time. Any changes that would affect the geometry or orientation of the model require that the GEOMETRY module be run again.

The preceding variables can be adjusted to calculate alternative PNPs. For example, if BUMPER calculated a PNP = 99% for a given Space Station configuration and orientation for 10 years in orbit, and it was necessary to calculate the effect of doubling the time in orbit, then the following technique should be used:

- a. Divided PNP by 100 to convert it to a decimal fraction.
 $\text{PNP} = 99\% / 100 = 0.99$
- b. Take the natural logarithm of the fractional form of PNP.
 $\ln(0.99) = -0.0101 = -\lambda T$
- c. Divide $-\lambda T$ by 10 years to calculate the λ term.
 $-\lambda = -0.0101 / 10 = -0.00101$
- d. Multiply the $-\lambda$ term by 20 years.
 $-\lambda T = -0.00101 * 20 = -0.0201$
- e. Calculate a new PNP based on the new time in orbit.
 $\text{PNP}' = 100\% * e^{-0.0201} = 98.01\%$

An example of this type of sensitivity study is shown in figure 4.1-1. This calculation is valid only if all other variables remain constant during the extended time in orbit. If the environment, surface area, or configuration is expected to change during that time, the technique in section 4.2 must be used. The previously described method can be used to study the effects of changes in either flux, time, or surface area with just a minimum of computing resources. (Note: doubling the value in step c produces an equivalent result by effectively doubling the time in orbit.)

4.2 STUDIES REQUIRING RERUNNING SOLUTION SEQUENCE

Studying changes in the Space Station's orientation, as shown in figure 4.2-1, requires a unique GEOMETRY analysis for each orientation. This requires a unique universal file with the nodal coordinates defined appropriately. SUPERTAB allows easy modification of the reference coordinate system, simplifying this task to a few keystrokes. The actual method used will depend on the model generation system selected by the user.

Sensitivity studies of changes in wall configuration require running GEOMETRY once to create either a debris or meteoroid data base and running RESPONSE once for each new wall configuration. BUMPER is then run for each unique response data base.

Sensitivity studies on the effects of changes to either the flux or the penetration function will require modification of either the RESPONSE or BUMPER modules, respectively. This will require careful study of section 7.0 and appendix C or D. If the changes in the flux are not constant, then the flux equation must be changed in the appropriate BUMPER subroutine. Changes in the penetration function must be handled in a similar manner, by modifying the appropriate equations in the various subroutines of RESPONSE.

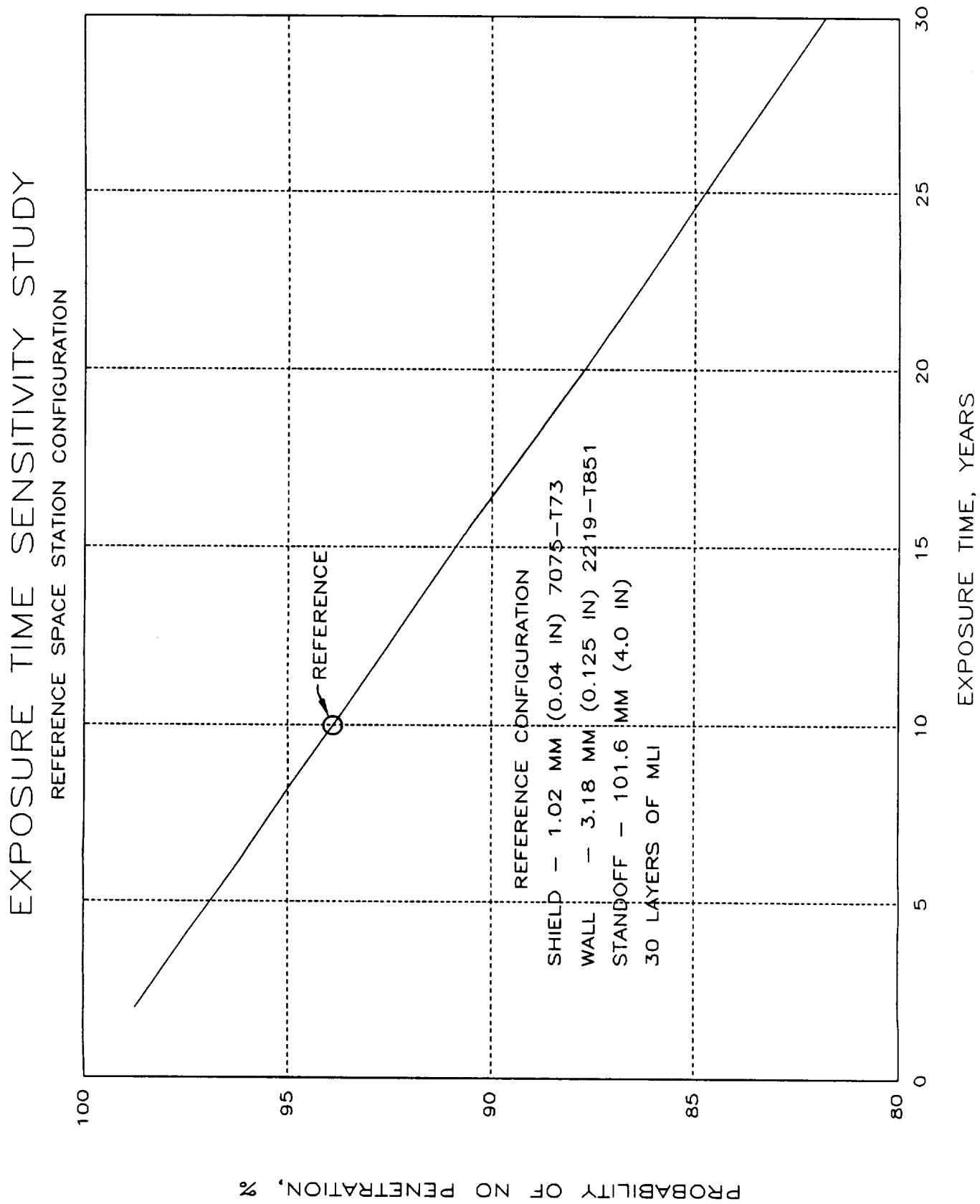


Figure 4.1-1. Results of Orbital Time Sensitivity Study

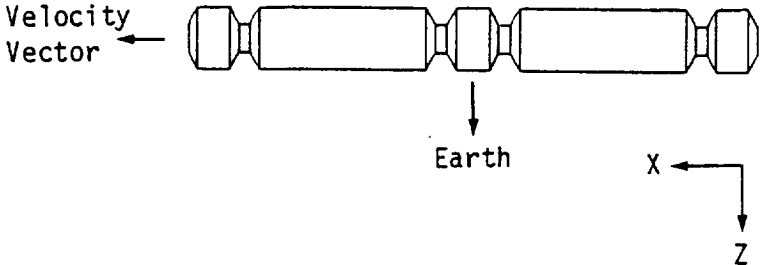
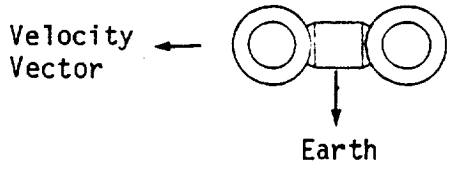
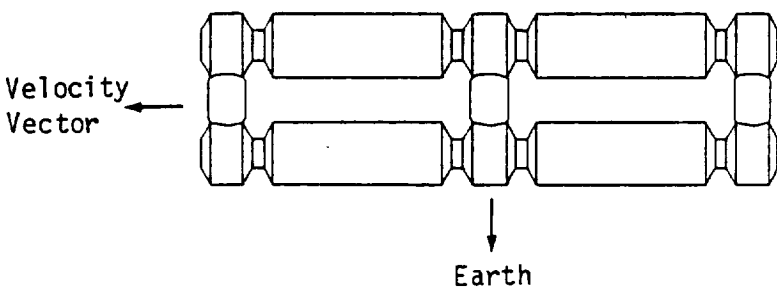
Orientations	Probability of No Penetration
<p>Reference Configurations</p> 	93.4%
<p>Rotation 90° Around Z Axis</p> 	95.3%
<p>Rotation 90° Around X Axis</p> 	87.8%

Figure 4.2-1. Results of Orientation Sensitivity Study

5.0 GENERATING CONTOUR PLOTS

The data for two types of contour plots can be generated by BUMPER: (1) a threat contour plot and (2) a design contour plot. Nevertheless, the actual method of displaying the data generated by BUMPER depends on the computing and graphics resources available to user.

5.1 THREAT CONTOUR PLOTS

The threat contour plots are generated as a postprocessing feature of SUPERTAB. The last question BUMPER asks during execution is whether the user wants to write out a SUPERTAB input file. If the user answers "YES" BUMPER writes out a file containing the probability of penetration per surface area for each element in the model. An example of a threat contour plot is shown in figure 5.1-1. Figure 5.1-2 shows a sample SUPERTAB input file written by BUMPER for generating the plot in figure 5.1-1. An explanation of the format used is shown in appendix A.

Use of preprocessors other than SUPERTAB will require either the developing a translator to change the data format or modifying subroutine SUPER in BUMPER.

5.2 DESIGN CONTOUR PLOTS

Generating design contours is a more involved process. The range of input variables must be defined by determining a minimum, maximum, and increment for shield thickness (T_s), vessel wall (T_b) thickness, and spacing (S) between the two. Examples of inputs to generate this data are given in section 6.0. The program CONTOUR then loops through the parameters, calculating the PNP for each configuration and writing T_s , T_b , S , PNP to the summary file. Figure 5.2-1 shows a sample design contour plot, and figure 6.4-2 shows the associated CONTOUR output file used to generate such a plot.

The actual program used to produce the contour plot is a user-supplied item. During the contract a Boeing-developed isotherm graphics program was used but any contouring package should work. Care should be taken when selecting a package. Most packages use a linear interpolation to determine the contours. If this is the case the log of the PNP should be plotted versus shield and vessel wall thickness because it more closely varies linearly.

SDRC--I-DEAS 2.5B: Output Display

19-JAN-86

LOAD CASE: 1

MIN: +0.000E+00 MAX: +5.796E-04

SPACE STATION

PROBABILITY OF PENETRATION (%) PER SQ-METER

	1	2	3	4	5
5.0E-05	1.5E-04	2.5E-04	3.5E-04	4.5E-04	

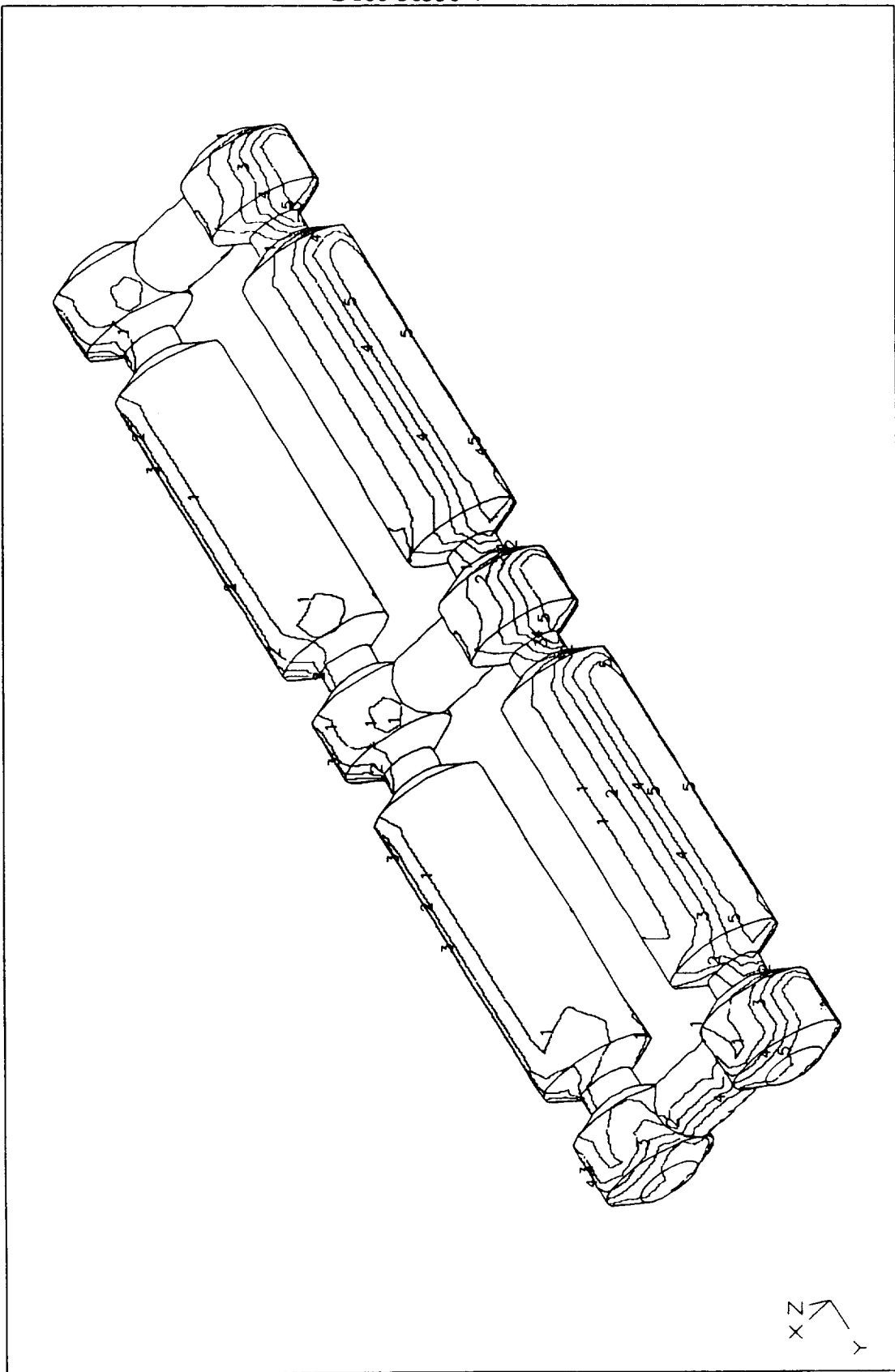


Figure 5.1-1. Threat Contour Plot -- Debris Environment

```

-1
56
MAN-MADE ORBITAL DEBRIS ANALYSIS
PROBABILITY OF PENETRATION (%) PER SQ-METER
ONE
ONE
ONE
      1      0      1      1      2      1
      1      1      1
0.00000E+00
  10000      1
0.11814E-05
  10005      1
0.30171E-04
  10020      1
0.21931E-02
  10025      1
0.21931E-02
  10030      1
0.21931E-02
  10035      1
0.21931E-02
  10040      1
0.21931E-02
      .      .
      .      .
      .      .
      .      .
      10045      1
      85780      1
0.18242E-02
      85790      1
0.18242E-02
-1

```

Figure 5.1-2. SUPERTAB Postprocessing Input for Threat Contout Plot

HABITABLE PORTIONS OF SPACE STATION PROBABILITY OF NO PENETRATION FOR REFERENCE CONFIGURATION

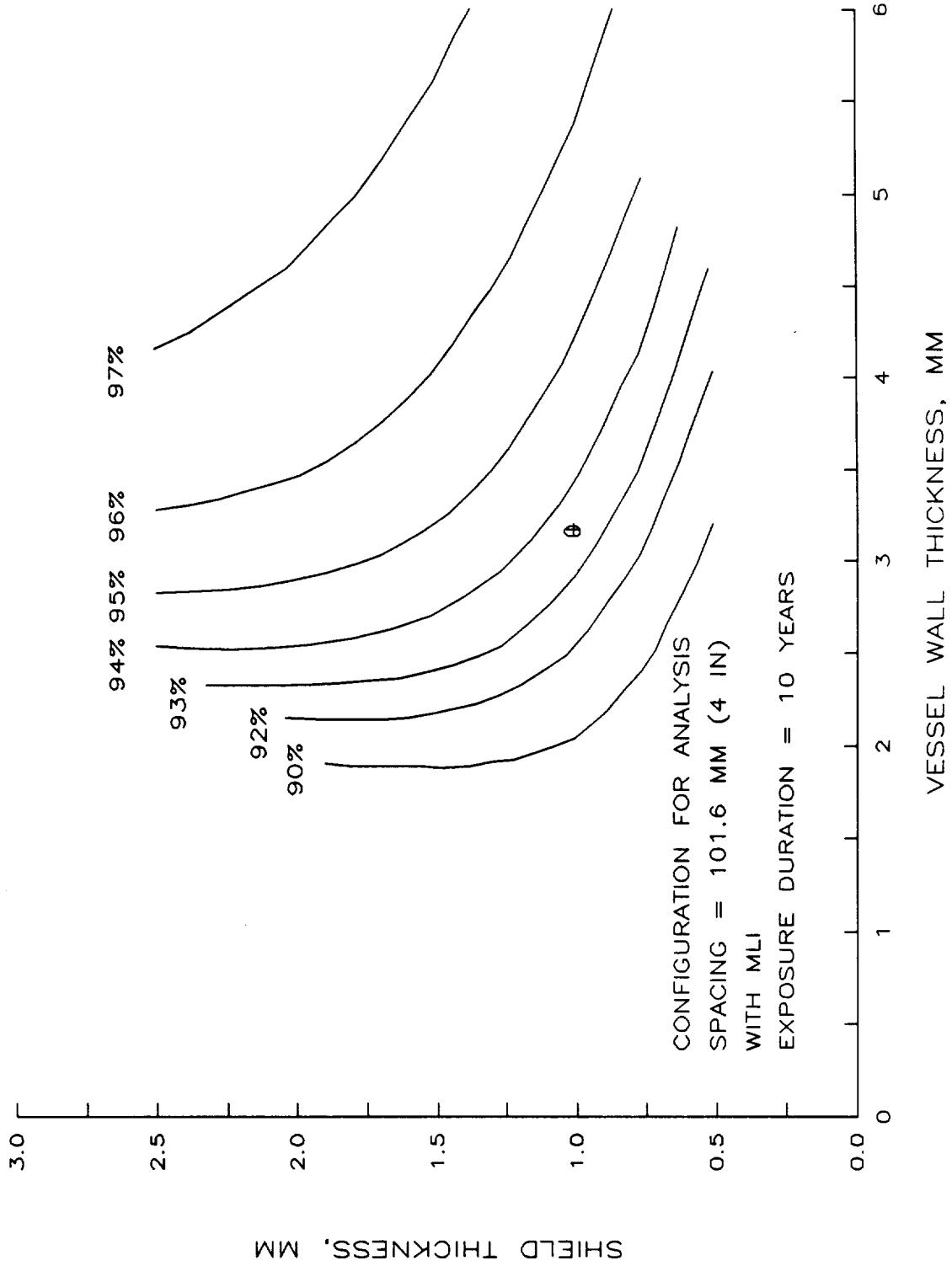


Figure 5.2-1. Design Contour Plot – Debris Environment

This section shows a sample of the execution of each module within BUMPER; a sample of the summary output file, and an explanation of each question as well as typical replies. Items in bold font are typed in by the user either to run a module or in response to questions.

6.1 GEOMETRY

Figure 6.1-1 shows the questions asked by GEOMETRY Version 2.51. The first question asks whether the geometry data base will be for debris or meteoroid analysis with the default being debris.

The second question asks for the name of the file containing the geometry data in a SUPERTAB-compatible Universal File Format. A carriage return defaults the file name to "STATION.UNI."

The third question asks for the data base file name. GEOMETRY will create this file and write out in a binary format the geometry data base. A carriage return defaults the data base filename to "STATION.GEM."

For debris analysis the fourth and final question asks for the number of uniform threats. The default is 45, which means that GEOMETRY will examine the model in 45 equal steps from +90.0 deg to -90.0 deg (every 4 deg) on either side of the X-axis or velocity vector, as shown in figure 6.1-2. GEOMETRY writes out a message as each threat case is completed and then indicates that it has successfully completed its function by writing out the name of the data base file.

If the desired analysis type was meteoroids, then the fourth question is about the number of uniform meteoroid threats. Figure 6.1-3 shows an example of a GEOMETRY run for meteoroid analysis. Figure 6.1-4 shows how a hemisphere is divided into approximately equal segments to generate a uniform series of meteoroid threats.

6.2 RESPONSE

Figure 6.2-1 shows the questions asked by RESPONSE Version 2.0 and gives typical responses. The first question asked, is if the threat to analyze is debris or meteoroids. The default is debris which means an aluminum projectile will be assumed. If meteoroids are chosen, an icy meteoroid projectile with a density of 0.5 g/c^3 will be assumed.

The second question asks for the wall property output file name. This summary file, shown in figure 6.2-2, records the analysis type and wall configuration used.

The third question asks for the response data base output file name. This is the binary data base contains the critical projectile diameter values as a function of PID, impact angle, and impact velocity. This file is then used by BUMPER to determine PNP.

RUN GEOMETRY

GEOMETRY VER 2.51

ANALYSIS TYPE ?

1-DEBRIS <CR>

2-METEORIDS

ANSWER 1 OR 2 >1

SUPERTAB UNIVERSAL FILENAME (CR=STATION.UNI) >STATION.UNI

OUTPUT FILENAME (CR=STATION.GEM) >EXAMPLE.GEM

NUMBER OF UNIFORM DEBRIS THREATS (CR=45) >45

THREAT CASE 1 COMPLETED

THREAT CASE 2 COMPLETED

THREAT CASE 3 COMPLETED

. . .
 . . .
 . . .
 . . .

THREAT CASE 42 COMPLETED

THREAT CASE 43 COMPLETED

THREAT CASE 44 COMPLETED

THREAT CASE 45 COMPLETED

OUTPUT LOCATED IN FILE EXAMPLE.GEM

\$

Figure 6.1-1. Example of GEOMETRY Run – Debris

D180-30550-4

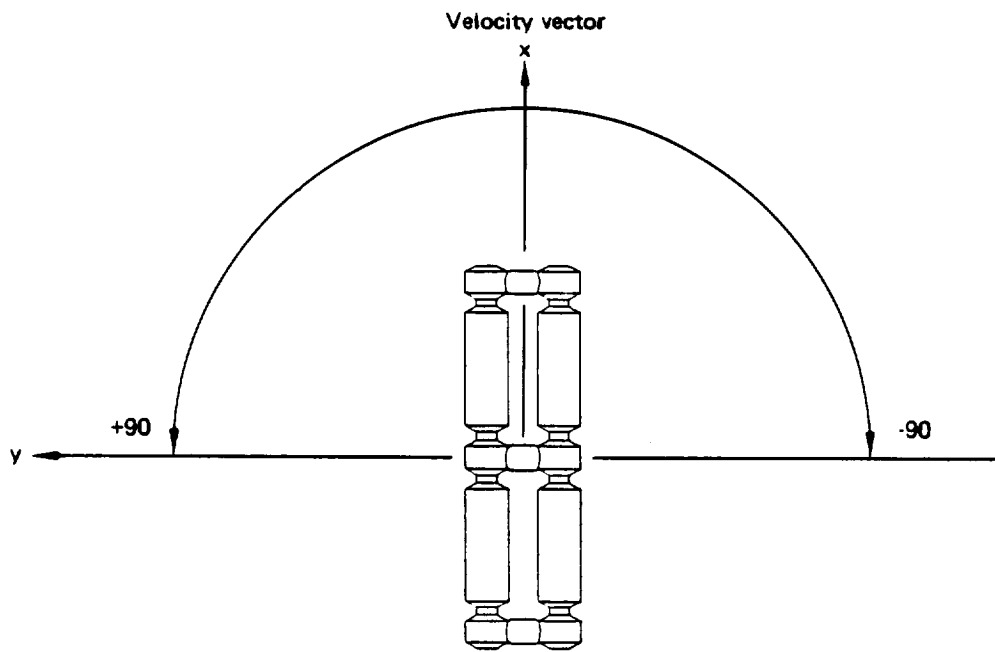


Figure 6.1-2. Debris Threat Generation

RUN GEOMETRY

GEOMETRY VER 2.51

ANALYSIS TYPE ?

1-DEBRIS <CR>

2-METEORIDS

ANSWER 1 OR 2 >2

SUPERTAB UNIVERSAL FILENAME (CR=STATION.UNI) >TUBES.UNI

OUTPUT FILENAME (CR=STATION.GEM) >EXAMPLE-2.GEM

NUMBER OF UNIFORM METEOROID THREATS ?

1 - 84

2 - 146<CR>

3 - 232

4 - 329

ANSWER (1-4) >2

THREAT CASE 1 COMPLETED

THREAT CASE 2 COMPLETED

THREAT CASE 3 COMPLETED

. . . .

. . . .

. . . .

THREAT CASE 143 COMPLETED

THREAT CASE 144 COMPLETED

THREAT CASE 145 COMPLETED

THREAT CASE 146 COMPLETED

OUTPUT LOCATED IN FILE EXAMPLE-2.GEM

\$

Figure 6.1-3. Example of GEOMETRY Run — Meteoroids

- Element coordinates selected to provide elements of equal area.
- Threat angles θ_i and ϕ_j are measured to centroid of curved surface element.
- All combinations of θ_i and ϕ_j are equally likely.

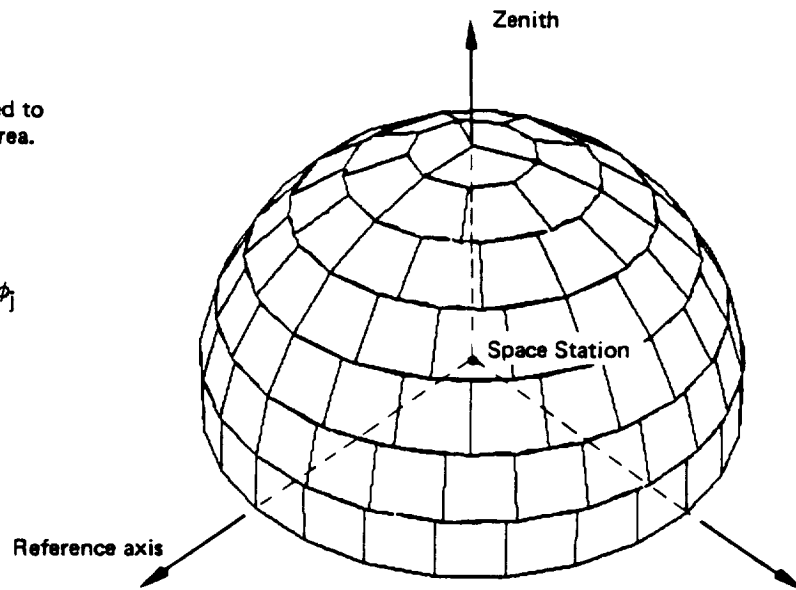


Figure 6.1-4. Meteoroid Threat Generation

RUN RESPONSE

RESPONSE VER 2.0

Last Update 5/25/87

ANALYSIS TYPE ?

1-DEBRIS <CR>

2-METEOROIDS

ANSWER 1 OR 2 : 1

OUTPUT FILENAME FOR WALL PROPERTIES <CR>=RESPONSE.SUM : RESPONSE.SUM

OUTPUT FILENAME FOR RESPONSE TABLES <CR>=STATION.RSP : EXAMPLE.RSP

INPUT IN METRIC OR ENGLISH UNITS <CR>=METRIC : METRIC

PROPERTY ID NUMBER = 1

CONFIGURATION TYPE

1- SINGLE PLATE

2- DOUBLE PLATE <CR>

ANSWER (1 or 2) : 2

PENETRATION FUNCTION

1-ORIGINAL <CR>

2-PEN4

3-REGRESSION

ANSWER (1-3) : 3

SHIELD MATERIAL

1- 2024-T4

2- 2219-T87

3- 6061-T6

4- 7075-T6

SELECT MATERIAL NUMBER <CR>=1 : 3

SHIELD THICKNESS (CM) = : 0.160

(continued on next page)

Figure 6.2-1. Example of RESPONSE Run

(continued from previous page)

VESSEL WALL MATERIAL

- 1- 2024-T4
- 2- 2219-T87
- 3- 6061-T6
- 4- 7075-T6

SELECT MATERIAL NUMBER <CR>=1 : 2

VESSEL WALL THICKNESS (CM) = : 0.3175

SHIELD STAND-OFF (CM) = : 10.160

INCLUDE 30 LAYERS OF MLI BETWEEN PLATES <CR>=YES : YES

DO YOU WISH TO RUN ANOTHER CASE ? <CR>=YES : NO

Figure 6.2-1. Example of RESPONSE Run (Continued)

RESPONSE VER 2.0
MAN-MADE ORBITAL DEBRIS ANALYSIS
PROPERTY ID 1
DOUBLE PLATE
REGRESSION PENETRATION FUNCTION
SHIELD MATERIAL = 6061-T6
SHIELD THICKNESS (CM) = 0.1600
VESSEL WALL MATERIAL = 2219-T87
VESSEL WALL THICKNESS (CM) = 0.3175
SHIELD STAND-OFF (CM) = 10.1600
30 LAYERS OF MLI BETWEEN SHIELD AND VESSEL WALL

\$

Figure 6.2-2. RESPONSE Summary File

The fourth question determines the length units to be used for input-output, either English (inches) or metric (centimeters).

(The code loops through the next series of questions generating data bases for each unique wall configuration.)

The fifth question determines the configuration type, either a single plate wall construction or a dual plate consisting of a shield (bumper) and pressure wall, the latter is the default.

The sixth question is for double plate analysis only and determines the penetration function to be used. These functions are described in more detail both in reference 1 and in section 7.4.6. The recommended penetration function for Space Station type applications is regression.

The next series of questions establish the material and plate thickness for the shield and vessel (pressure) wall, the standoff distance between the two plates (for dual plate configuration), and whether or not 30 layers of MLI are to be included between the plates. For single plate analysis only, the vessel wall material and thickness is read in.

The last question asks whether the user wants to run another case (configuration). If the answer is "NO," then the code writes the data base and summary output file; otherwise it increases the property identification number (PID) in increments of one and resumes asking questions about the wall configuration. The default answers on subsequent wall configuration questions is the same as the answer on the previous configuration. The user must trace the number of PIDs and where they appear in the model.

6.3 BUMPER

Figure 6.3-1 shows the questions asked by BUMPER Version 4.0 and gives typical responses.

The first question asks for the name of the summary file, an example of which is shown in figure 6.3-2. As with RESPONSE, this summary output file can be used by the user as a record of the filenames, analysis, and results.

The second question establishes the analysis type, either debris or meteoroids. This determines which flux equation is used within BUMPER.

The third question asks for the exposure time (orbit time) in years. The default is 10 years.

The fourth question asks for the operating altitude of the spacecraft in kilometers. When the altitude varies with time (which is the case with Space Station), the recommended altitude is the antilogarithm of the sum of the logarithms (base 10) of the average yearly altitude. An example of the orbital altitude calculation is shown in figure 6.3-3. If this information is not available, then the average or nominal altitude should be used.

RUN BUMPER

BUMPER VER 4.0

Last Update 5/25/87

SUMMARY OUTPUT FILENAME (CR=BUMPER.SUM)>

ANALYSIS TYPE ?

1-DEBRIS <CR>

2-METEORIDS

ANSWER 1 OR 2 >1

SPACE STATION EXPOSURE TIME (YEARS) <CR=10.0> : 10.0

OPERATING ALTITUDE (400-500 km) <CR=500> : 500.0

THE PROBABILITY OF NO PENETRATION WILL BE CALCULATED
FOR SPECIFIC RANGES OF ELEMENT IDS

INPUT THE STARTING AND ENDING ELEMENT ID FOR EACH RANGE

ENTER D <CR> OR <CR> WHEN DONE

RANGE 1

STARTING ELEMENT ID : 1

ENDING ELEMENT ID : 10000

RANGE 2

STARTING ELEMENT ID :

GEOMETRY OUTPUT FILENAME ? <CR=STATION.GEM> >EXAMPLE.GEM

RESPONSE OUTPUT FILENAME ? <CR=STATION.RSP> >EXAMPLE.RSP

(continued on next page)

Figure 6.3-1. Example of BUMPER Run

(continued from previous page)

THREAT CASE 1 COMPLETED
 THREAT CASE 2 COMPLETED
 THREAT CASE 3 COMPLETED

. . .
 . . .
 . . .

THREAT CASE 42 COMPLETED
 THREAT CASE 43 COMPLETED
 THREAT CASE 44 COMPLETED
 THREAT CASE 45 COMPLETED

RANGE	STARTING EID	ENDING EID	PNP %	Aeff sq-m
1	1	10000	96.11964	152.60001

TOTAL PROBABILITY OF NO PENETRATION (%) = 96.11964

TOTAL EFFECTIVE AREA (SQ-M) = 152.60001

CREATE A SUPERTAB INPUT FILE FOR CONTOUR PLOTS ? <CR=YES> >NO

\$

Figure 6.3-1. Example of BUMPER Run (Continued)

BUMPER VER 4.0

Last Update 5/25/87

MAN-MADE ORBITAL DEBRIS ANALYSIS
 SPACECRAFT EXPOSURE TIME (YEARS) = 10.00
 OPERATING ALTITUDE (km) = 500.00
 GEOMETRY OUTPUT FILE = EXAMPLE.GEM

RESPONSE OUTPUT FILE = EXAMPLE.RSP

RANGE	STARTING EID	ENDING EID	PNP %	Aeff sq-m
1	1	10000	96.11964	152.60001

TOTAL PRROBABILITY OF NO PENETRATION (%) = 96.11964

TOTAL EFFECTIVE AREA (SQ-M) = 152.60001

\$

Figure 6.3-2. BUMPER Summary File

Year	Altitude	Log ₁₀ (Altitude)
1	420	2.6232
2	430	2.6335
3	445	2.6484
4	460	2.6628
5	440	2.6435
		$\Sigma = 13.2114$

$$\text{Average} = \frac{13.2114}{5} = 2.6423$$

$$\text{Altitude} = 10^{(2.6423)} = 438.8 \text{ km}$$

Figure 6.3-3. Example of Orbital Altitude Calculation.

The next series of questions define the starting and ending element IDs for each range (SSCE). The example in figure 6.3-2 is for the entire model; therefore, the range input is from 1 to 10,000, which encompasses the entire range of element IDs. Figure 3.4-1 shows an example of a summary output file with many SSCEs defined.

The next two questions identify the appropriate data base files generated by GEOMETRY and RESPONSE. These files are in a binary format for speed of execution and compact storage.

The last question asks whether the user wishes to create a SUPERTAB-compatible file containing the probability of penetration per element surface area data for each element. Figure 5.1-2 shows an example of this file. Figure 5.1-1 shows how this data may be displayed.

6.4 CONTOUR

Figure 6.4-1 shows the typical questions asked by the CONTOUR and gives the typical responses.

The first question asks for the summary file. This file will contain the user-supplied input and the results of the analysis. With the default being "CONTOUR.SUM." Figure 6.4-2 shows an example of this file.

The second question asks for the analysis type, with the default being debris.

The third question asks for the exposure time, with a default value of 10 years.

The fourth question asks for the operating altitude. The recommended value is the same as outlined in section 6.3 for BUMPER. The default value is 500 km.

The next question asks for the element ID range for the calculations. As opposed to BUMPER, CONTOUR allows only one range. No default values are given.

The sixth question asks for the length units to be for input and output. The default is metric.

The next question asks for property ID, with the default being 1. Only one PID is allowed per analysis.

Type of wall configuration is then input, with the default being double plate. If double plate is chosen, then the penetration function is requested, with the default being the original one. The recommended function for Space Station type structure is the regression.

The next questions deal with the wall materials and minimum, maximum, and increment in thickness. No default is supplied. For

```
#####

CONTOUR Version 2.0

last update 6/5/87

#####

SUMMARY OUTPUT FILENAME (CR=CONTOUR.SUM) :

ANALYSIS TYPE ?
  1-DEBRIS <CR>
  2-METEOROIDS
ANSWER 1 OR 2 :

SPACECRAFT EXPOSURE TIME (YEARS) <CR=10.0> :

OPERATING ALTITUDE ( 400-500 km ) <CR=500> : 411

THE PROBABILITY OF NO PENETRATION WILL BE CALCULATED FOR A SPECIFIC RANGE
OF ELEMENT IDS INPUT THE STARTING AND ENDING ELEMENT ID FOR EACH RANGE

STARTING ELEMENT ID : 10000

ENDING ELEMENT ID   : 20000

INPUT IN METRIC OR ENGLISH UNITS <CR>=METRIC : ENGLISH

PROPERTY ID - <CR=1> :

CONFIGURATION TYPE
  1- SINGLE PLATE
  2- DOUBLE PLATE <CR>
ANSWER (1 or 2) :

PENETRATION FUNCTION
  1-ORIGINAL <CR>
  2-PEN4
  3-REGRESSION
ANSWER (1-3) :

SHIELD MATERIAL
  1- 2024-T4
  2- 2219-T87
  3- 6061-T6
  4- 7075-T6
  5- INCONEL
SELECT MATERIAL NUMBER <CR>=1 : 3
```

Figure 6.4-1 Example of CONTOUR Run

```

MINIMUM SHIELD THICKNESS (IN) - : .01
MAXIMUM SHIELD THICKNESS (IN) - : .1
INCREMENT SHIELD THICKNESS (IN) - : .01

VESSEL WALL MATERIAL
1- 2024-T4
2- 2219-T87
3- 6061-T6
4- 7075-T6
5- INCONEL
SELECT MATERIAL NUMBER <CR>-1 : 2

MINIMUM VESSEL WALL THICKNESS (IN) - : .075
MAXIMUM VESSEL WALL THICKNESS (IN) - : .25
INCREMENT VESSEL WALL THICKNESS (IN) - : .025

THE NUMBER OF POTENTIAL CASES IS      80
DO YOU WISH TO CONTINUE ? <CR=YES> :

SHIELD STAND-OFF (IN) - : 4.325

INCLUDE 30 LAYERS OF MLI BETWEEN PLATES <CR>-Y :

GEOMETRY OUTPUT FILENAME ? <CR=STATION.GEM> >

  1 0.0100 4.3250 0.0750    98.11780
  2 0.0200 4.3250 0.0750    98.78834
  3 0.0300 4.3250 0.0750    99.05371
  . . . . .
  . . . . .
  . . . . .
  . . . . .
  . . . . .
 66 0.0900 4.3250 0.2500    99.83081
 67 0.1000 4.3250 0.2500    99.83659

```

Figure 6.4-1 Example of CONTOUR Run (Continued)

D180-30550-4

#####

CONTOUR Version 2.0

last update 6/5/87

#####

MAN-MADE ORBITAL DEBRIS ANALYSIS

SPACECRAFT EXPOSURE TIME (YEARS) - 10.00

OPERATING ALTITUDE (km) - 411.00

STARTING ELEMENT ID - 10000

ENDING ELEMENT ID - 20000

PROPERTY ID - 1

DOUBLE PLATE ANALYSIS

ORIGINAL PENETRATION FUNCTION

SHIELD MATERIAL - 6061-T6

MINIMUM MAXIMUM AND INCREMENT SHIELD THICKNESS (IN) -
0.0100 0.1000 0.0100

VESSEL WALL MATERIAL - 2219-T87

MINIMUM MAXIMUM AND INCREMENT VESSEL WALL THICKNESS (IN) -
0.0750 0.2500 0.0250

SHIELD STAND-OFF (IN) - 4.32500

30 LAYERS OF MULTI-LAYER INSULATION INCLUDED
GEOMETRY OUTPUT FILE - STATION.GEM

1	0.0100	4.3250	0.0750	98.11780
2	0.0200	4.3250	0.0750	98.78834
3	0.0300	4.3250	0.0750	99.05371
.
.
.
.
66	0.0900	4.3250	0.2500	99.83081
67	0.1000	4.3250	0.2500	99.83651

Figure 6.4-2 Sample CONTOUR Summary File

single plate structure, only the vessel wall questions are asked.

Once all the wall parameters are input, the number of potential configurations is calculated and displayed. If the user agrees with the number the analysis continues by inputting "YES" or a <CR>. If the number is incorrect, then the wall inputs may be repeated by inputting "NO."

Next, for double plate structure, the shield standoff and whether or not to include MLI is input.

The last question asks for the geometry data base filename. This was the file created by the GEOMETRY code. The default is "STATION.GEM."

As each configuration is analyzed the results are written to both the screen and the summary file. An example of the summary file is shown in figure 6.4-2.

6.5 RPLOT

RPLOT reads the binary response data base and produces a formatted user-defined output file containing the critical diameter data for each PID and impact velocity for 0-, 15-, 30-, 45-, and 60-deg impact angles. Figure 6.5-1 shows the typical questions asked by the program. As can be seen, the questions are limited to the input and output filenames. Figure 6.5-2 shows a typical RPLOT output file.

```
RUN RPLOT
$$$$$$$$$$$$$$$$$$$$
      RPLOT VER 1.1
$$$$$$$$$$$$$$$$$$$$
RESPONSE OUTPUT FILENAME <CR-STATION.RSP> :STATION.RSP
OUTPUT FILENAME <CR-RPLOT.DAT> : RPLOT.DAT
OUTPUT LOCATED IN RPLOT.DAT
```

Figure 6.5-1 Example of RPLOT Run

D180-30550-4

PID	Vi	0	15	30	45	60
1	0.2500	1.0366	1.0725	1.1973	1.4640	2.0707
1	0.5000	0.6506	0.6731	0.7515	0.9195	1.3009
1	0.7500	0.5035	0.5216	0.5815	0.7123	1.0092
1	1.0000	0.4236	0.4388	0.4891	0.5990	0.8474
.
.
.
.
1	16.2500	0.6979	0.7040	0.7235	0.7611	0.8300
1	16.5000	0.6953	0.7013	0.7207	0.7582	0.8268
1	16.7500	0.6927	0.6987	0.7180	0.7554	0.8237
1	17.0000	0.6901	0.6961	0.7154	0.7526	0.8207
2	0.2500	0.9562	0.9899	1.1039	1.3541	1.9153
2	0.5000	0.6166	0.6382	0.7131	0.8716	1.2327
2	0.7500	0.4818	0.4986	0.5564	0.6808	0.9628
2	1.0000	0.4062	0.4204	0.4690	0.5738	0.8113
.
.
.
.
	16.2500	0.6243	0.6297	0.6471	0.6807	0.7424
2	16.5000	0.6219	0.6273	0.6446	0.6782	0.7395
2	16.7500	0.6195	0.6249	0.6422	0.6756	0.7368
2	17.0000	0.6172	0.6226	0.6398	0.6731	0.7340

Figure 6.5-2 Sample RPLLOT Output File

D180-30550-4
7.0 APPLICATION OF THEORY

7.1 HISTORY

In the past determining the probability of no penetration (PNP) of spacecraft subject to meteoroid impact involved the Poisson's model as described in equation 1.

$$PNP = e^{-FAT} \quad (1)$$

where PNP is the probability of no penetration, F is the flux of the design particle, A is the spacecraft area, and T is the exposure time.

Previously, because spacecraft had small areas and short exposure times, the shielding required using this analysis was not significant. Also, the effect of man-made orbital debris was not previously considered.

When using the preceding analysis for the design of the Space Station, the required shielding for man-made debris became excessive. Exact meanings of the terms in this equation were also confusing. What area is used, surface, projected, effective, or some factor of either? How is a design particle selected?

This lead to the development of a more rigorous technique, outlined in reference 1, appendix G. The benefit of this more detailed analysis is that it accounts for the actual geometry of the spacecraft, its orientation, and all of the impact characteristics. This leads to more acceptable shielding requirements in terms of weight without compromising the required reliability of the spacecraft.

7.2 DETERMINING THE PROBABILITY OF NO PENETRATION

The analysis technique in reference 1, appendix G breaks the spacecraft into a series of flat plate elements much like a FEM. The threat environment (both debris and meteoroids) is also broken into finite cases, each for which a probability can be assigned. Each element is then evaluated for each threat. The PNP for a specific element is given by

$$PNP = \text{EXP} \left[- [A \cdot T \cdot \sum_{i=1}^{NT} F(D_c(v_i, \beta_i)) \cdot \cos(\beta_i) \cdot p(\alpha_i, \theta_i) \cdot I_s] \right] \quad (2)$$

where A is the elements surface area, T is the exposure time, NT is the number of threats, β is the angle between the plates normal and the threat vector (the impact angle), $p(\alpha_i, \theta_i)$ is the probability of the threat occurring, $F(D_c(V_i, \beta_{ij}))$ is the flux of the particle that penetrates the plate for the given impact velocity and impact angle, and I_s is a indicator equal to 1 if the element is exposed to the threat and 0 if it is not.

To determine the PNP for a range of elements the following equation is used

$$PNP = \text{EXP} \left[- \left[T * \sum_{j=1}^{NELM} A_j * \sum_{i=1}^{NT} F(D_c(v_{ij}, \beta_{ij})) * \cos(\beta_{ij}) * \rho(\alpha_i, \theta_i) * I_s \right] \right]^{(3)}$$

where NELM is the number of elements.

The preceding equations show that several items of data are required for each element. These data can be broken into logical groups. The first group consists the data specific to the threat and the spacecraft geometry. This consists of the threat data (impact velocity and the probability of the threat occurring) and the spacecraft interaction with the threat (is the element exposed and the cosine of the impact angle). These data are calculated by the GEOMETRY computer program.

The second group consists of data specific to the penetration resistance of the spacecraft wall (the diameter that just penetrates for specific impact conditions). These data are calculated by the computer program RESPONSE.

This information is brought together along with the exposure time and element range of interest to perform the PNP calculation in the BUMPER computer program.

Breaking the PNP calculation into several distinct parts has many advantages. The most expensive part of the analysis, in terms of computer runtime, is creating the threat and spacecraft geometry interaction data base. This data base need only be created once for each spacecraft geometry and orientation. Trade studies involving changes in the wall configuration or spacecraft exposure time then use this common data base. Breaking down the analysis also produces modular computer codes that are more adaptable to modifications and future additions.

A detailed discussion of each of these computer codes follows:

7.3 GEOMETRY VERSION 2.51

GEOMETRY produces the geometry data base used by the BUMPER code. Specifically, it produces information about the threat (polar angles, impact velocity and probability), the finite elements in the model (element ID, property ID, surface area), and the interaction between the model and the threat (list of exposed elements and their impact angle cosines for each threat). The information is stored in a user-defined binary file.

The environment is modeled as a series of distinct threat cases as outlined in section 7 of reference 1. The spacecraft geometry is input through the use of a FEM. The model is limited to triangular elements whose nodes must be in the same coordinate system. In addition, the X-axis of the this coordinate system must be parallel

to the spacecraft's velocity vector and the Z-axis must be parallel to the Earth radius vector pointing away from the Earth. This is shown in figure 2.2-1. The model information must be stored in a file using the SUPERTAB Universal File Format. Other formats could be used but would require modification of the code.

Figure 7.3-1 shows a simplified flow chart of the computer program. Each line item in the figure will be discussed in more detail in the following sections, but it should be noted that no attempt will be made to describe the code in complete detail. The code is well documented internally, and appendix B contains a complete listing.

The code consists of two main areas, as shown in figure 7.3-1. The first consists of the global calculations, which need only be calculated once. The remainder of the code is specific to determining which elements are exposed to a given threat. This portion of the code can be further broken down into two areas, the first being the elimination of the elements that do not face the threat (back side elements) and the second being the elimination of elements shielded from the threat by other elements.

7.3.1 Write Header

This portion of the code writes the program header to the screen. It also reads in the analysis type (meteoroid or debris), the name of the file containing the FEM, and the name of the output file.

7.3.2 Calculate the Threat Data

The threat data for meteoroids or debris is calculated here. These data consist of the polar angles describing the threat direction, the impact velocity associated with the threat, and the probability of the threat occurring. The user specifies the number of threats to be evaluated.

The debris threat is defined by JSC 20001. The code uses the relative velocity distribution and derives from it the threat angle distribution. The velocity distribution is contained in an external file DEB.VEL. The format for the velocity distribution file is contained in the code listing. The relationship between relative velocity and threat angle are shown graphically in figure 7.3-2.

The meteoroid threat is defined by NASA SP-8013. For simplification, a constant inertial velocity of 20 km/s for all meteoroids is assumed. An isotropic threat is also assumed; therefore, all directions have equal likelihood of occurring. The spacecraft motion is accounted for by increasing or decreasing the probability appropriately (see sec. 2 of ref. 1). Earth shadowing is accounted for by ignoring the directions that lay below the horizon. For meteoroid analysis, a 500 km spacecraft altitude is assumed. For a more detailed discussion of the threat, see sections 2 and 7 of reference 1.

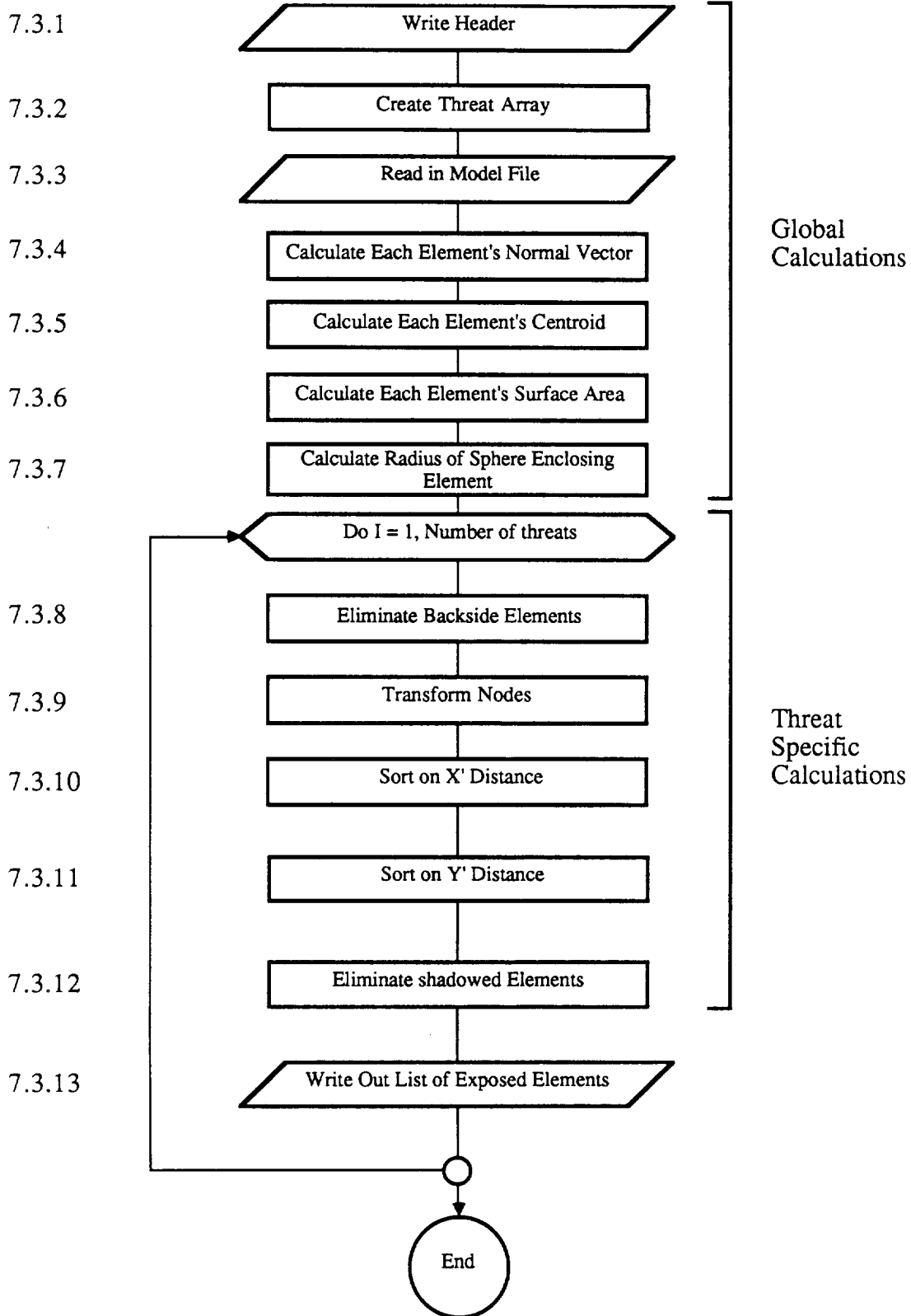


Figure 7.3-1. GEOMETRY Flow Chart (Version 2.0)

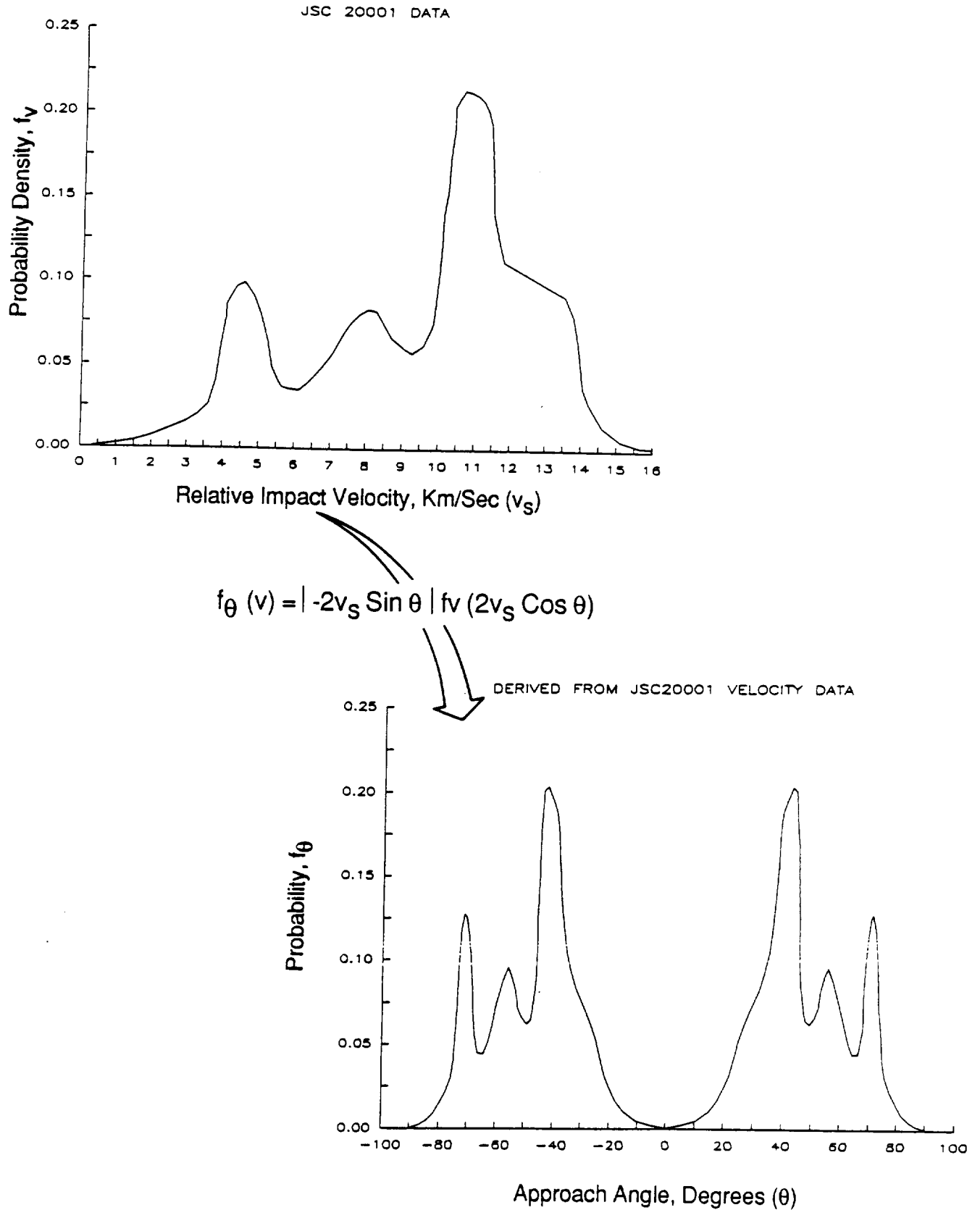


Figure 7.3-2. Transformation of Velocity Distribution to Approach Angle Distribution

7.3.3 Read in the Model File

The model file is read in and stored in global arrays for further use. One benefit of the code is the use of common blocks contained in external files, allowing easy modification of memory requirements.

7.3.4 Calculate the Normal Vectors

Each element's normal vector is calculated in the global coordinate system. The model builder must take care to ensure the correct orientation. The code uses nodes in order 1, 2, 3 and the right-hand rule to assign the normal vector direction.

7.3.5 Calculate the Centroid Locations

Each element's centroid is calculated in the global coordinate system by averaging the three nodal locations.

7.3.6 Calculate the Surface Areas

Each element's surface area is calculated and saved for output.

7.3.7 Calculate the Radius of Sphere Enclosing the Element

The radius of the sphere that just encloses each element is calculated and saved. This information is used later to evaluate the elements for shadowing.

7.3.8 Eliminate Back Side Elements

For each threat the elements having normal vectors pointing away from the threat are eliminated to prevent impacts on the element back faces. This is done by calculating the cosine of the angle between the element's normal vector and a vector pointing toward the threat direction. This is also the cosine of the impact angle and is saved for output later if the element is exposed. Those elements with negative cosines are termed "back side elements" and are removed from further consideration for that specific threat.

7.3.9 Transform the Nodal Coordinates and Centroid Locations

Those elements not eliminated as back side elements now have their nodal coordinates and centroid locations transformed into a coordinate system where the X-axis is parallel to the threat direction. The coordinates are then projected onto a plane orthogonal to the threat direction located at the centroid of the element. This is shown in figure 7.3-3.

7.3.10 Sort on the Transformed X-Dimension

The potentially exposed elements (those not eliminated as back side) are sorted in descending order on the distance from the origin to the elements centroid in the transformed X-dimension shown in figure 7.3-3. The quick-sort algorithm is used.

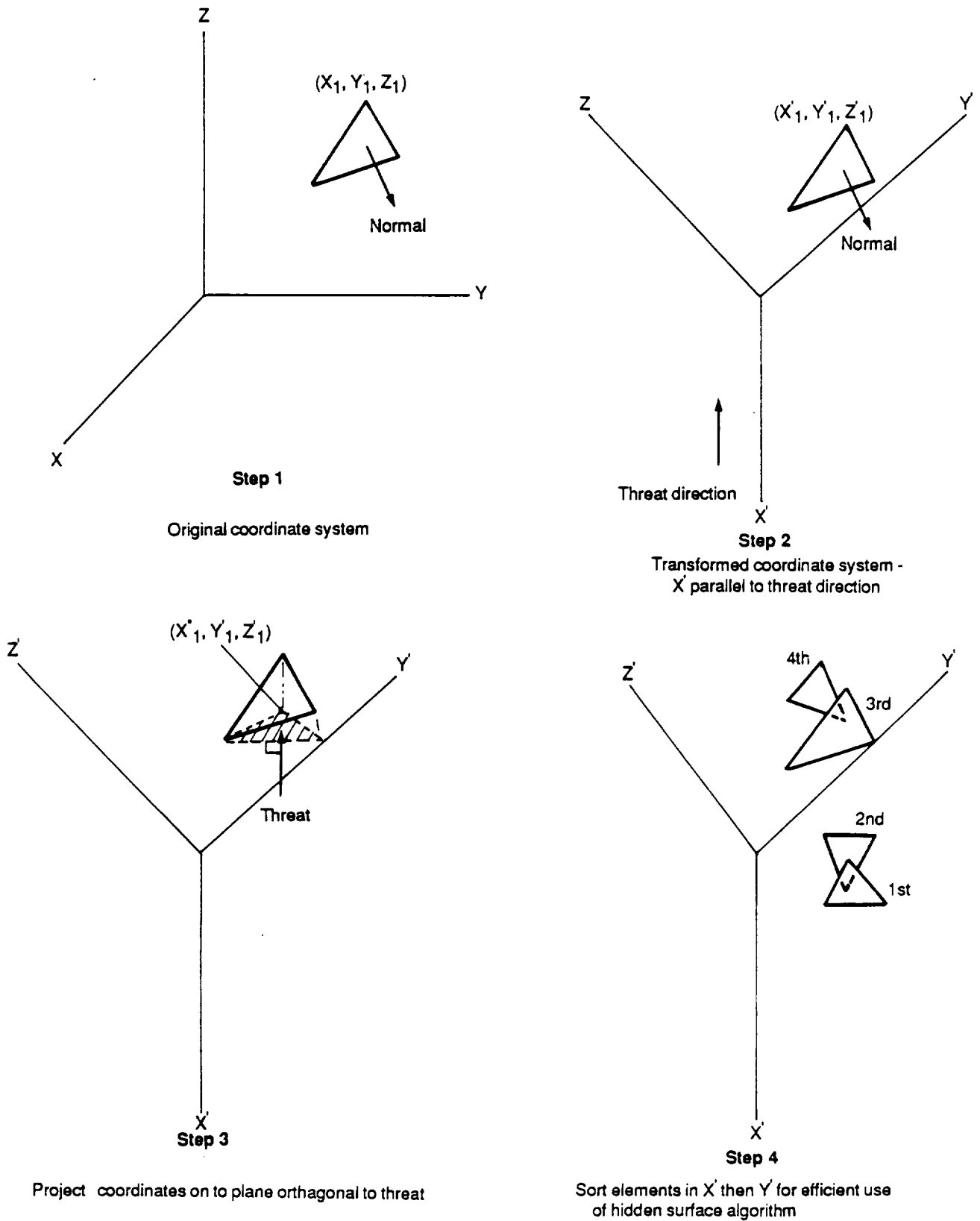


Figure 7.3-3. Transformation of Nodes and Sorting of Elements

7.3.11 Sort on the Transformed Y-Dimension

The potentially exposed elements are sorted in descending order on the distance from the origin to the element's centroid in the transformed Y-dimension shown in figure 7.3-3. The quick-sort algorithm again is used.

7.3.12 Eliminate the Shadowed Elements

An element is in the shadow of another element only if its centroid lies behind the shadowing element and within the projected boundaries of the shadowing element. This is done with the sorted lists of elements in the transformed X- and Y-dimensions. The potentially shadowed element must lie below the shadowing element in the sorted transformed X-list. Only those elements whose centroids lay within a distance of the radius enclosing the shadowing element in the transformed Y-dimension are considered. This is done using the sorted transformed y list. If an element passes this test, a more accurate test is performed to determine if it lies within the projected boundaries. This portion of the code is complex and is best understood by following an example through the shadowing subroutine. This will not be done in this document, but the subroutine is well documented internally.

7.3.13 Output List of Exposed Elements and Associated Impact Angles

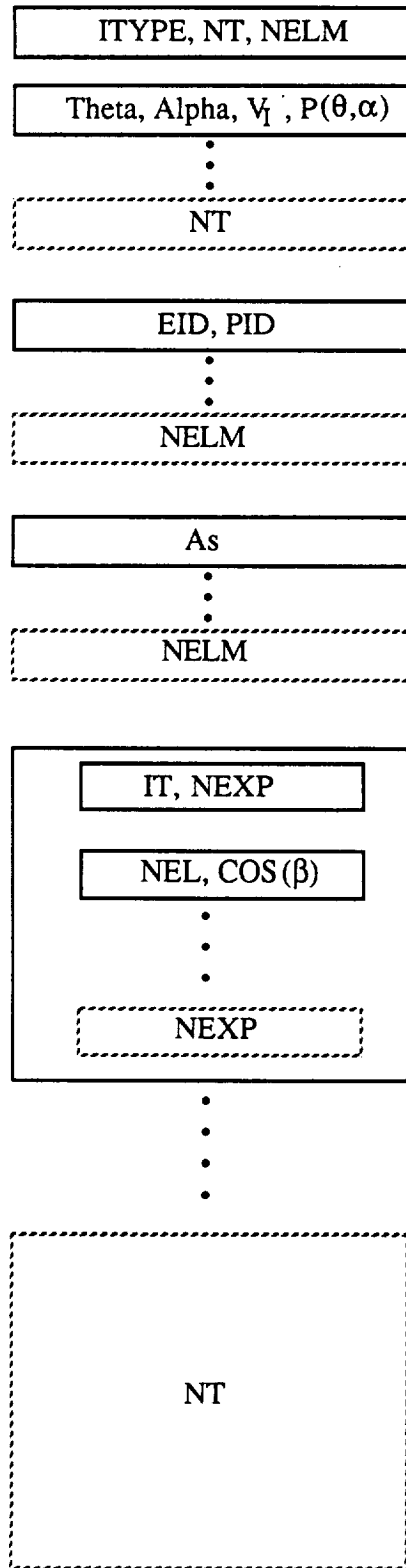
For each threat case a list of the exposed elements and their associated impact angle cosines is output. Additional outputs are the analysis type, the threat data, element surface area, element ID, and property ID. All this information is contained in binary format in the user-specified output file. Figure 7.3-4 shows the structure of this file.

7.4 RESPONSE VERSION 2.0

RESPONSE generates the critical diameter lookup tables for use by the BUMPER code. These tables list the critical diameter as a function of property ID (each finite element has been assigned one when the model was built), impact angle, and impact velocity. The information is saved in a user-defined file in binary format. The code is limited to aluminum structures, and specifically for double plate structures the shield thickness must be between 10% and 50% of the vessel wall thickness.

A simplified flow chart of the code is shown in figure 7.4-1. Each line item in the flow chart will be discussed in more detail in the following sections, but no attempt will be made to explain all the details of the code. The source code is internally documented, and a full listing of it is contained in appendix C.

The code can be broken into two parts for discussion purposes. The first part consists of reading in the user-inputs. These consist of the type of penetration function to be used and various dimensions and materials. The second part deals with calculating the critical diameters (the diameter that just penetrates) as a function



Integer*2

IT = Threat case
 ITYPE = Analysis type (1 - Deb, 2 - Met)
 NT = Number of threat cases
 NELM = Number of elements
 NEL = Element number
 NEXP = Number of exposed elements for a particular threat case

Integer*4

EID = Element ID
 PID = Element Property ID

Real*4

As = Surface Area, m²
 COS(β) = Cosine of impact angle
 P(θ,α) = Probability of threat occurring
 V_I = Impact velocity for specific threat, km/s
 Theta, Alpha = Polar angles describing threat

Figure 7.3-4. Geometry Data Base File Structure.

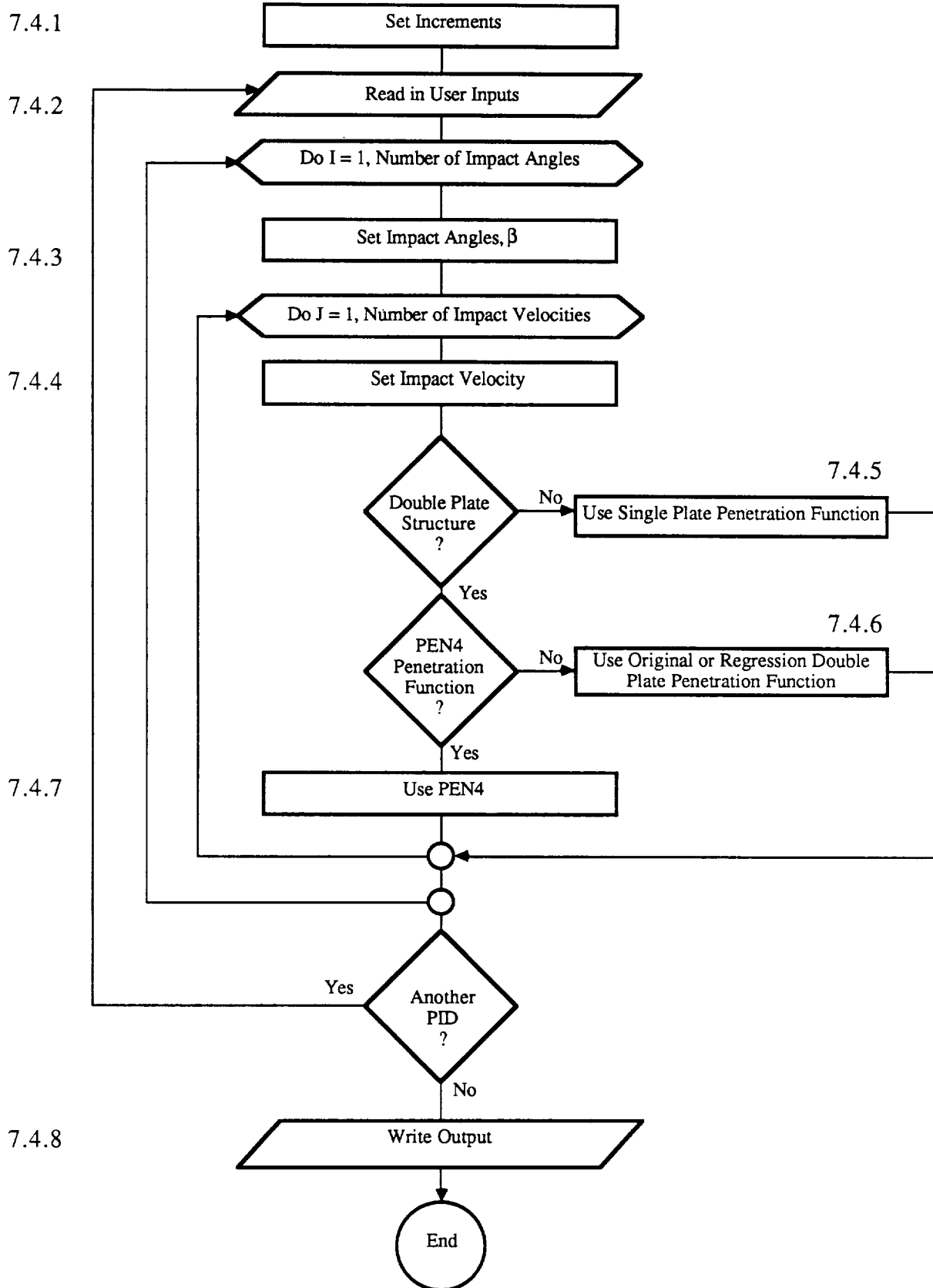


Figure 7.4-1. RESPONSE Flow Chart (Main Program.)

of impact angle and impact velocity. The results are output in the user-defined file.

7.4.1 Set Increments

The increments for impact angle and impact velocity are explicitly set in the code. For meteoroid analysis the impact angle is varied in 5-deg increments and the impact velocity in 1 km/s increments with a maximum impact velocity of 70 km/s. For man-made debris the impact angle is also varied in 5-deg increments, while the impact velocity is varied in 0.25 km/s increments with a maximum impact velocity of 17 km/s. These values can be changed but would require modification of the source code.

7.4.2 Read in User Inputs

The user-specified inputs are read in at this point. For the initial property ID they consist of the analysis type, output file names, and the type of input units (English or metric). Additionally, the type of wall configuration (single or double plate) is read in. For double plate structures three optional penetration functions are available. The user also specifies the plate thicknesses, the shield standoff, and whether or not multilayer insulation is included. For all additional property IDs the default values are those input for the previous property ID.

This portion of the code also reads in the material property file MAT.PRP. This file contains the physical property data for several common aluminum alloys. Additional aluminum alloys can be added. The format of the file is shown in figure 7.4-2.

7.4.3 Set the Impact Angle

The code sets the impact angle using the increments defined in section 7.4.1. Because data is limited for impact angles greater than 60 deg, the code considers angles greater than 60 deg to be equivalent to 60 deg.

7.4.4 Set the Impact Velocity

The impact velocity is set using the previously defined increments given in section 7.4.1.

7.4.5 Single Plate Penetration Function

The response of single plates subject to meteoroid or debris impact is modeled with the Schmidt-Holsapple crater volume equation, (sec. 4 and 7 of ref. 1). The effect of multilayer insulation is ignored. In addition spall is addressed by defining failure as hemispherical crater depths exceeding 70% of the plate thickness. A typical response curve is shown in figure 7.4-3.

7.4.6 Double Plate Penetration Function

As stated previously, the user has the option of three unique penetration functions for double plate structures. However, once one

Figure 7.4-2. Example of MAT.PRP File (Material Property File.)

Material Name	Density (lb/in ³)	Tensile Yield Strength (psi)	Tensile Ultimate Strength (psi)	Shear Strength (psi)	Wilkinson's ¹ Constant (km/s)	Speed of Sound (ft/s)	Shock ² Projectile Velocity (unitless)	Brinell Hardness
2024-T4	.100	37000.	54000.	29000.	.425	17024.	1.345	120.
2219-T87	.102	52000.	63000.	37000.	.270	16620.	1.345	130.
6061-T6	.098	27000.	42000.	35000.	.292	16630.	1.345	73.
7075-T6	.101	68000.	78000.	47000.	.345	16540.	1.345	150.

Fortran Format (A12, 8E12.5)

References:

- 1 "A Penetration Criterion For Double-Wall Structures Subject To Meteoroid Impact," J.P.D. Wilkinson, AIAA Journal, October 1969.
- 2 Empirical Material Constant for Aluminum.

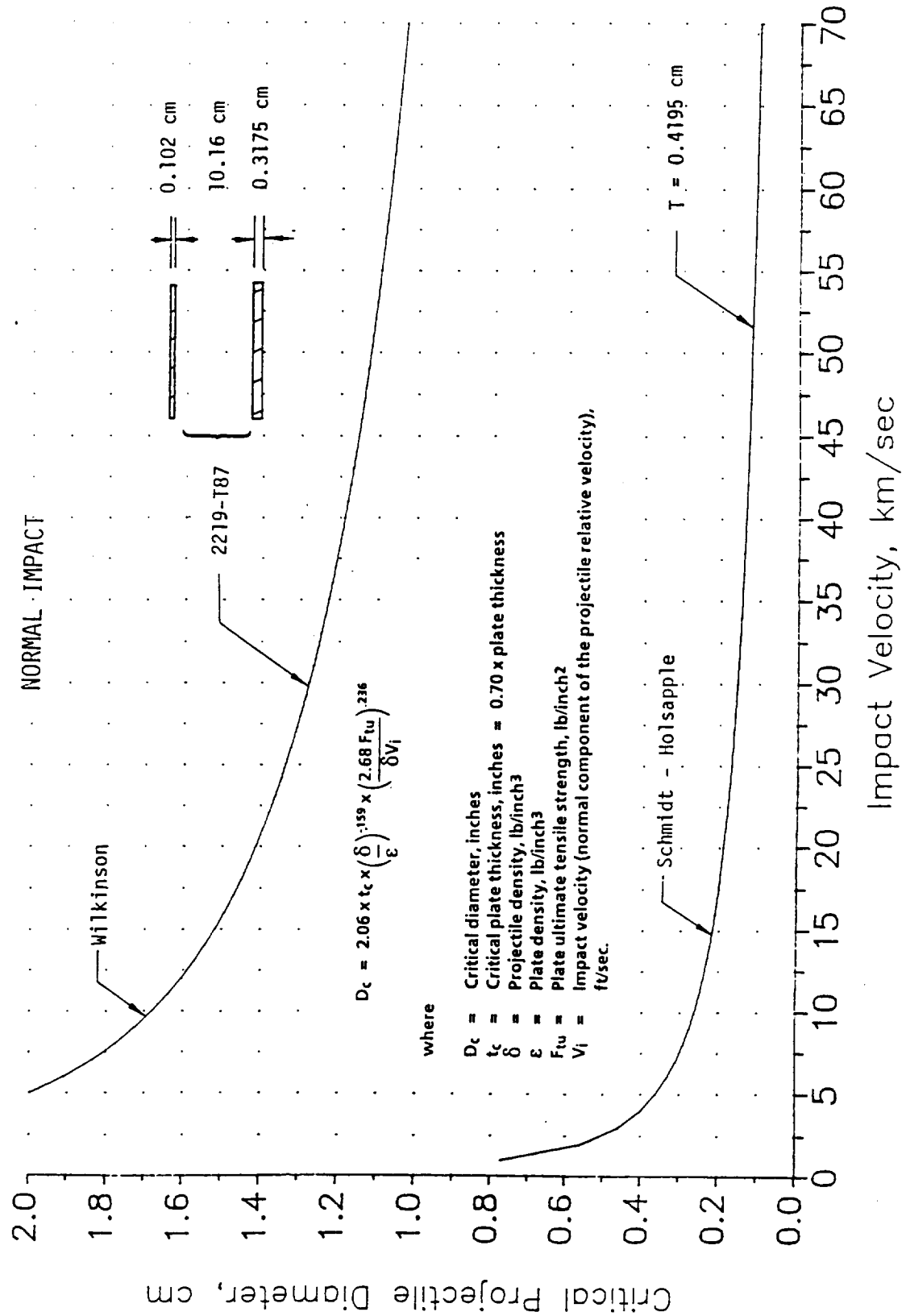


Figure 7.4-3. Comparison of Penetration Resistance to Meteoroids of Single Wall and Double Wall Structures

has been chosen, all following property IDs in that analysis session will use the same penetration function.

The first penetration function is titled the original. This is the function with which the majority of the contract work was performed. It consists of the ballistic portion of the original PEN4, the Burch modified Bristow equations (ref. 1) for the fragmenting regime, and the Wilkinson equations for the high-velocity regime. A typical penetration function using this approach is shown in figure 7.4-4. It should be noted that the transition between the various equations is only a function of where the equations intercept and not based on physical data.

The second penetration function is the updated version of PEN4 and is discussed below in section 7.4.7.

The third penetration function is titled regression. It uses the updated version of PEN4 for the ballistic regime, a regression equation for the fragmenting regime, and Wilkinson's method for the high-velocity regime. The regression equation represents the best curve fit of the test data developed under the contract. For a more detailed discussion, see ref. 1. The regression penetration function is recommended for use in Space Station type analysis. Figure 7.4-5 shows a typical penetration function using this technique.

7.4.7 PEN4

PEN4 is the Boeing-developed hypervelocity penetration computer program. Developed by the Boeing Aerospace hypervelocity impact group, it is intended to analyze impacts in the range of 0 to 8 km/s. It has been updated to more closely predict the penetration resistance of aluminum double plate structure impacted by aluminum projectiles. A typical penetration function is shown in figure 7.4-6. A more detailed discussion of PEN4 is contained in ref. 1. It should be noted that the code does not address the effect of multilayer insulation, and the results for angles other than normal impact are considered to be suspect.

7.4.8 Write Output

The response lookup tables are written in binary format to the user-defined output file. Additionally, a summary list of information on each property ID case run is written to the user-defined summary file. Figure 7.4-7 shows the structure of this file.

7.5 BUMPER VERSION 4.0

BUMPER determines the PNP of a spacecraft subject to man-made debris or meteoroid impact. The code also determines the effective area of the spacecraft. The code can perform these calculations for all elements in the spacecraft model together or for specific ranges of element IDs. Figure 7.5-1 shows a simplified flow chart of the computer code. Each line item will be discussed in further detail in the following sections, but no attempt will be made here to explain

IWALL TEST PROGRAM

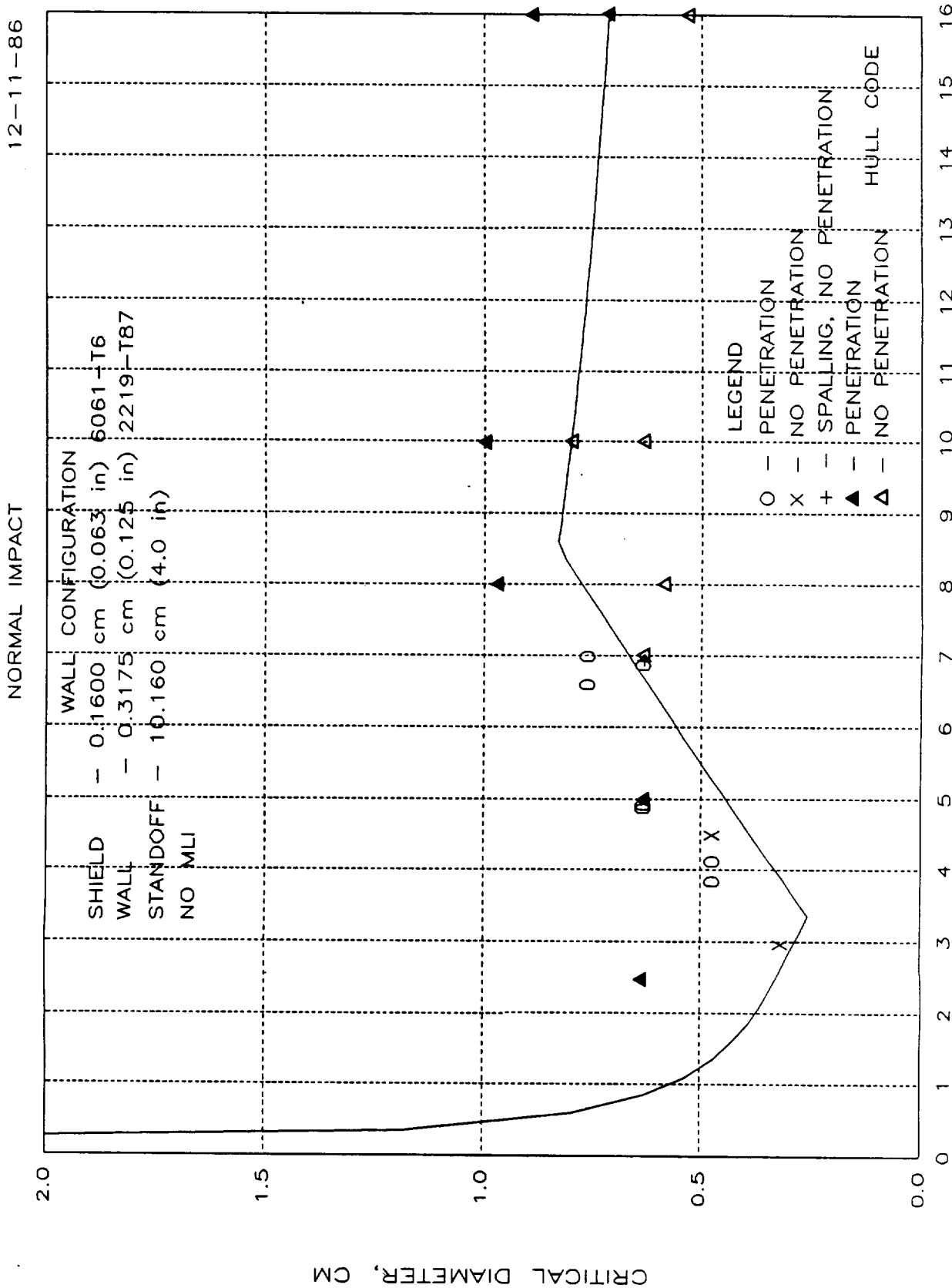


Figure 7.4-4. Penetration Function With Burch Equations

IWALL TEST PROGRAM

REGRESSION FUNCTION - NORMAL IMPACT

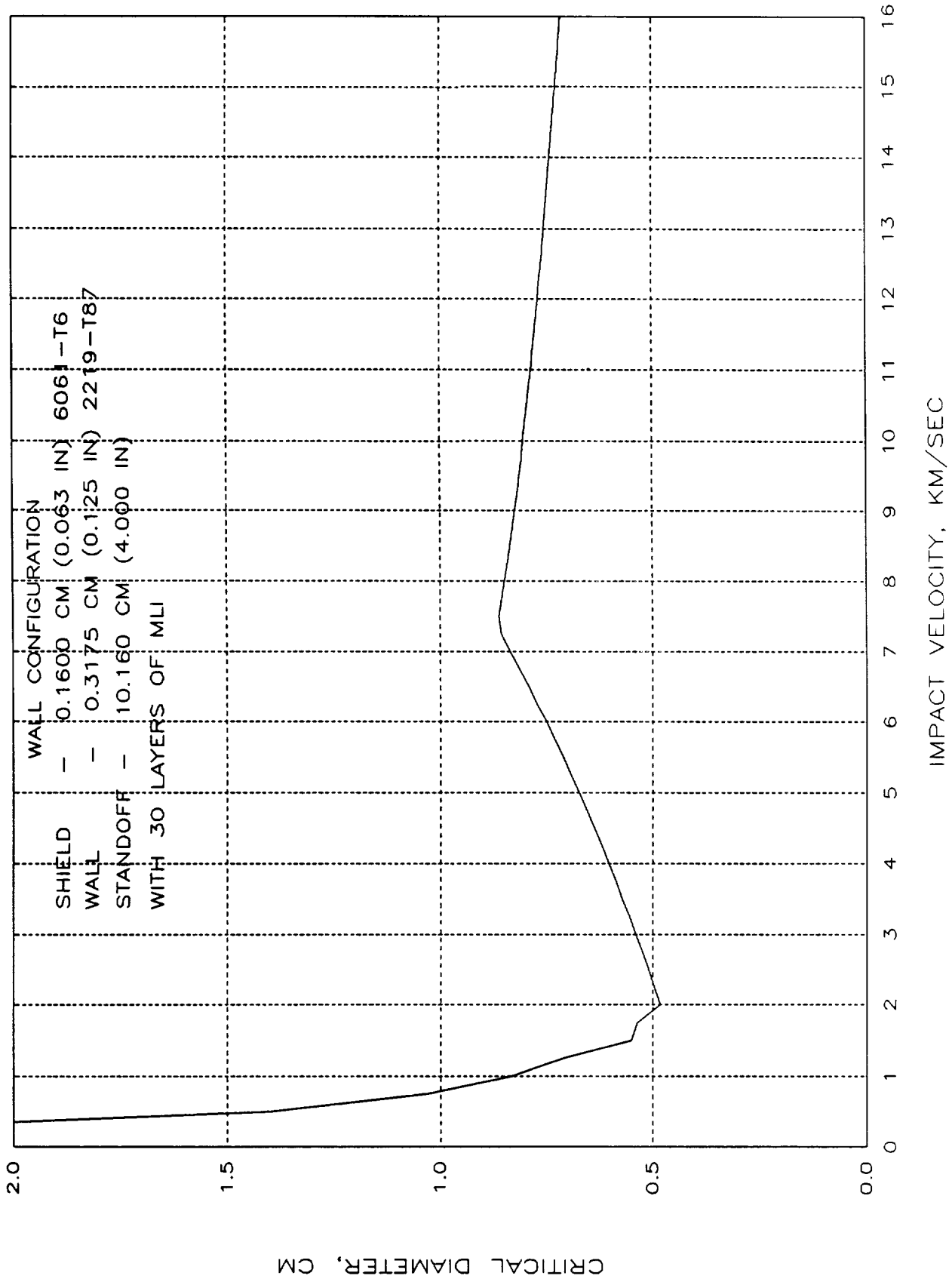


Figure 7.4-5. Regression Penetration Function

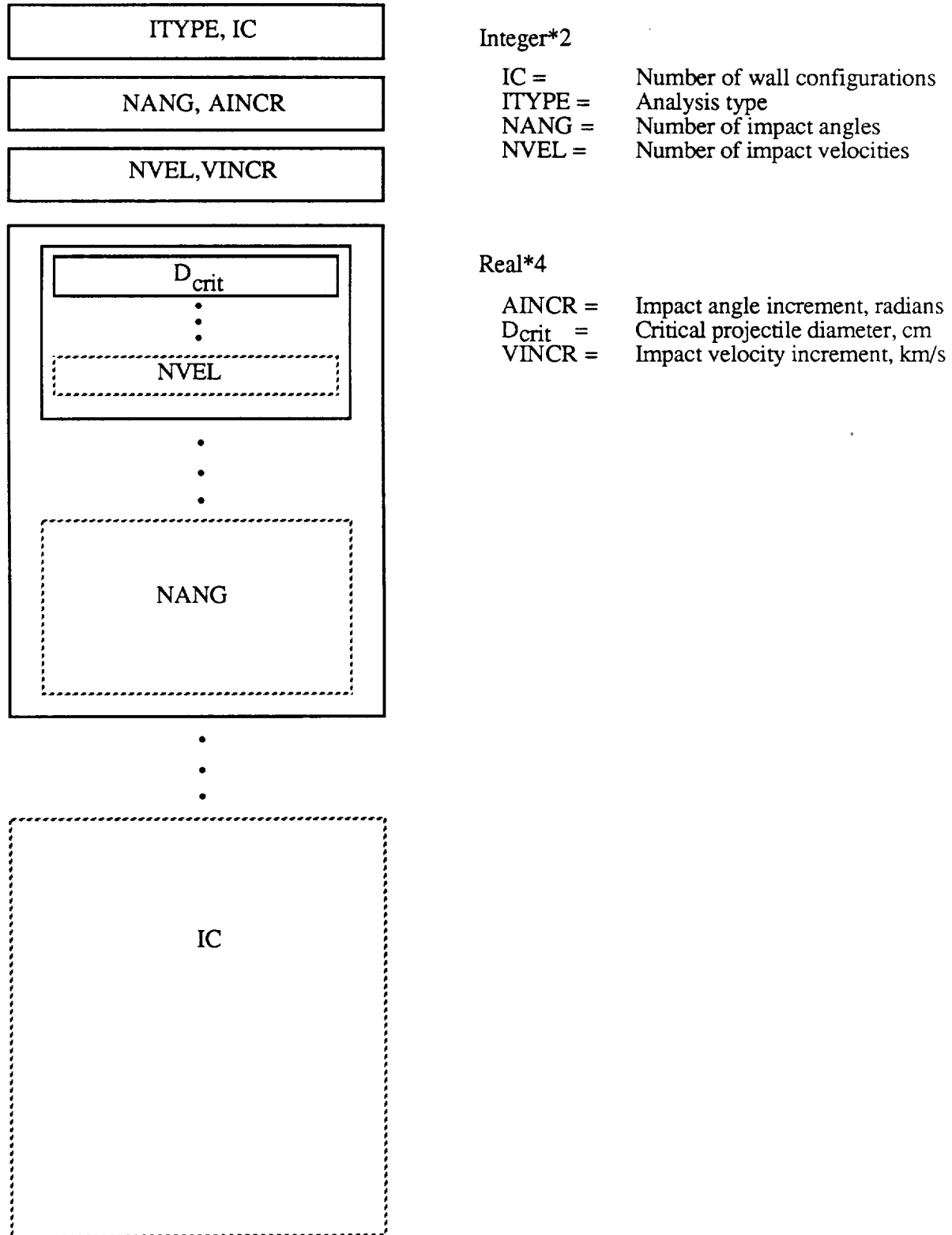


Figure 7.4-7. Response Data Base File Structure.

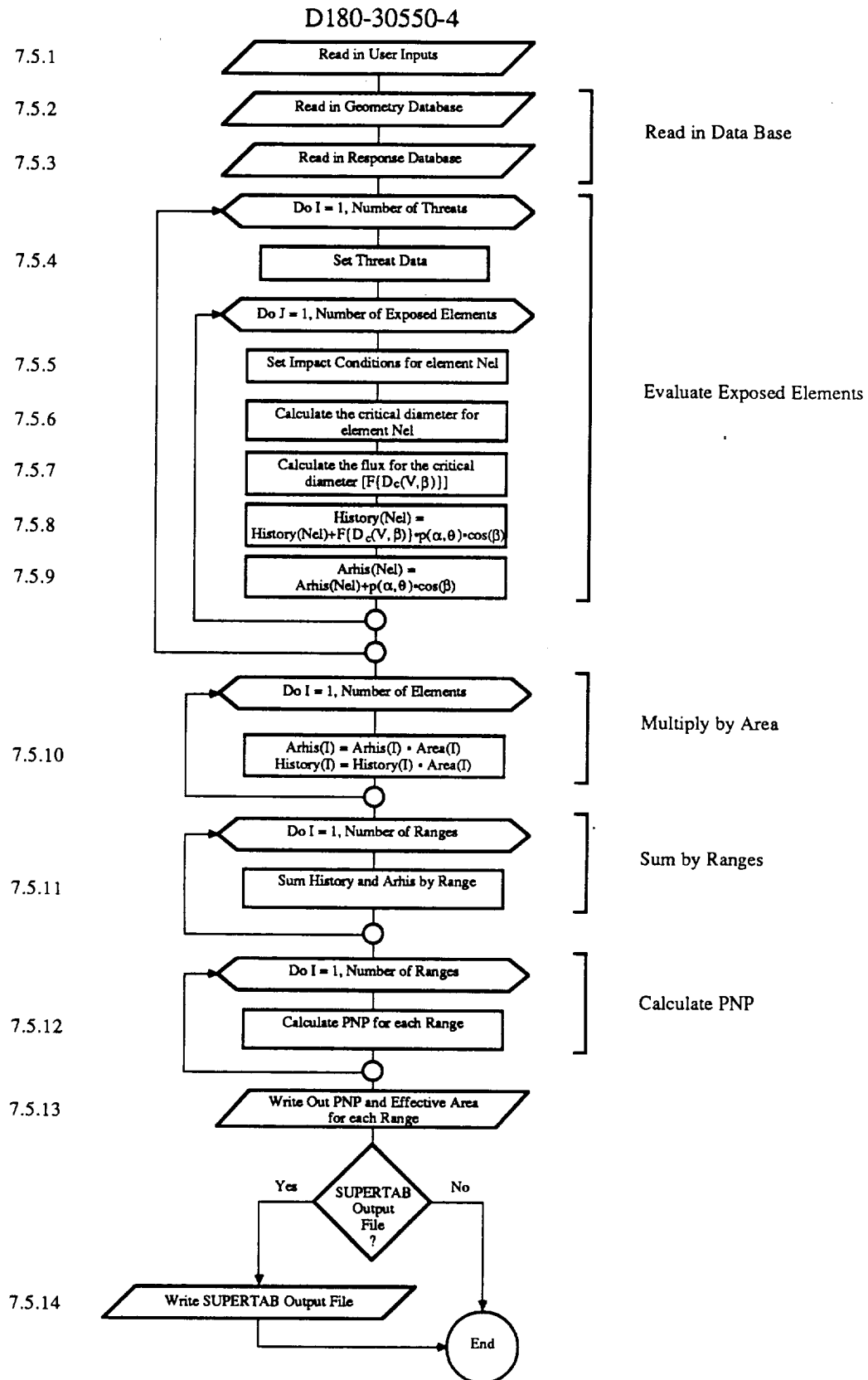


Figure 7.5-1. BUMPER Flow Chart.

the code in complete detail. The code is well documented internally, and a complete listing is contained in appendix D.

As shown in figure 7.5-1, the code can be broken into five areas for discussion purposes. The first consists of reading in the geometry and response data bases. Next, the finite threat cases are looped through, evaluating only the exposed elements for each one. This consists of determining the flux of the critical projectile and storing its sum in a global array. This global array is then multiplied by the element surface area array. The array is then summed up by element ID ranges. Finally, the PNP for each range is calculated.

7.5.1 Read User Inputs

The user inputs consist of the analysis to be performed (meteoroid or debris), the summary filename, the spacecraft exposure time, the spacecraft operating altitude, and the element ID ranges. This portion of the code also writes the screen header. All user input is output to the summary file.

7.5.2 Read in the Geometry Data Base

The geometry data base created by the GEOMETRY code and stored in a binary file is read in. The data base contains the information on the threat, each element's element identification number (EID) and PID, the element's surface area, and a list of the exposed elements along with their impact angle cosine for each threat.

7.5.3 Read in the Response Data Base

The wall penetration resistance data base created by the RESPONSE code and stored in a binary file is read. The data base contains the critical diameter lookup tables for each property ID as a function of impact velocity and impact angle.

7.5.4 Set the Threat Data

For each threat the probability of the threat occurring and the impact velocity is recovered from the geometry data base.

7.5.5 Set the Impact Conditions for Element Nel

For each threat there is a list of exposed elements in the geometry data base. Nel is a specific element exposed to a specific threat. The code loops through all exposed elements for all threats. The impact conditions consist of the impact angle cosine and the property ID for the element as well as the previously recovered impact velocity.

7.5.6 Calculate the Critical Diameter

The critical diameter for element Nel for the specific threat can now be calculated. The response data base is used through linear interpretation to determine the critical diameter. Using a lookup table approach as opposed to calculating the diameter directly allows a faster, more efficient code.

7.5.7 Calculate the Flux of the Critical Diameter

The flux of the critical diameter is calculated using NASA-specified flux equations. The meteoroid flux equations are found in JSC 30000. Gravity focusing of meteoroids is accounted for. Earth shadowing has already been accounted for in the GEOMETRY code. The man-made debris flux equations are found in JSC 20001. The log of the debris flux is assumed to vary linearly between 400 and 500 km. The flux given by JSC 20001 is multiplied by 4.0 to account for the difference in the Boeing and NASA definition of flux. No corrections are needed for the meteoroid flux. For a more detailed discussion, see appendix G of ref. 1.

$$7.5.8 \text{ HISTORY(NEL)} = \text{HISTORY(NEL)} + F(D_c(v_i, \beta_i)) * \cos(\beta_i) * \rho(\alpha_i, \theta_i)$$

The product of the flux, the threat probability, and the cosine of the impact angle for each element is summed in the History array for all elements. When all threats have been evaluated the sum in the History array for each element represents the summation part of equation 1.

$$7.5.9 \text{ ARHIS(NEL)} = \text{ARHIS(NEL)} + \cos(\beta_i) * \rho(\alpha_i, \theta_i)$$

The product of the threat probability and the cosine of the impact angle for each element is summed in the Arhis array for all threats. When all threats have been evaluated, this represents the effective area divided by the surface area for element Nel.

$$7.5.10 \text{ HISTORY(NEL)} = \text{HISTORY(NEL)} * \text{Area(NEL)},$$

$$\text{ARHIS(NEL)} = \text{ARHIS(NEL)} * \text{Area(NEL)}$$

Each element in the History and Arhis arrays is multiplied by its surface area. Each term in the History array now represents the average number of penetrating particles per unit time for a specific element. Each term in the Arhis array represents the effective area of a specific element.

7.5.11 Sum up the History and the Arhis Arrays by Ranges

The History array is summed up for the various user defined ranges. This summation represents the average number of penetrating particles per unit time for a specific range of elements (representing an SSCE, for example). The summation of the Arhis array represents the effective area for each range of elements.

7.5.12 Calculate PNP for Each Range

Given the spacecraft exposure time and the average number of penetrating particles per unit time for each range, the PNP for

each range can be calculated. A Poisson model, as shown in equation 2, is used.

7.5.13 Write PNP and Effective Area Out

The PNP and effective area for each range of element IDs as well as the total PNP and effective area, are written out to both the screen and the summary file.

7.5.14 SUPERTAB Output File

A SUPERTAB Universal File containing the probability of penetration per surface area for each element in the model can be produced. This allows for penetration contour plots on the model to be made.

7.6 CONTOUR VERSION 2.0

CONTOUR produces a data base that can be used with user-supplied software to produce design contour plots of PNP versus shield and vessel wall thickness for given shield standoff, element ID range, and property ID. The code is a modified version of the BUMPER code. The main modification being the incorporation of the RESPONSE code as a subroutine. Figure 7.6-1 shows a simplified flow chart of the code. Those items that are different from the previous discussion of the BUMPER code will be discussed in more detail in the following sections. The code is well documented internally, and a complete listing is contained in appendix E.

7.6.1 Read in User Inputs

The user inputs are the same as those in the BUMPER and RESPONSE codes. Additionally, the minimum, maximum, and increment are input for the shield and vessel wall thickness. The code is limited to one range of element IDs and one property ID. If, during the analysis, elements in the specified range are encountered with the incorrect property ID, then the code skips the element. A warning message is written to the summary file indicating how many elements with the incorrect property ID were encountered.

7.6.2 Set Vessel Wall Thickness

The vessel wall thickness (T_b) is set using the user-defined minimum, maximum, and increment.

7.6.3 Set Shield Thickness

The shield thickness (T_s) is set using the user-defined minimum, maximum, and increment.

7.6.4 Ratio OK

The ratio of the shield thickness to the combined shield and vessel wall thickness is calculated. This ratio is limited to a range of 0.10 to 0.50. This is due to limitations in the various

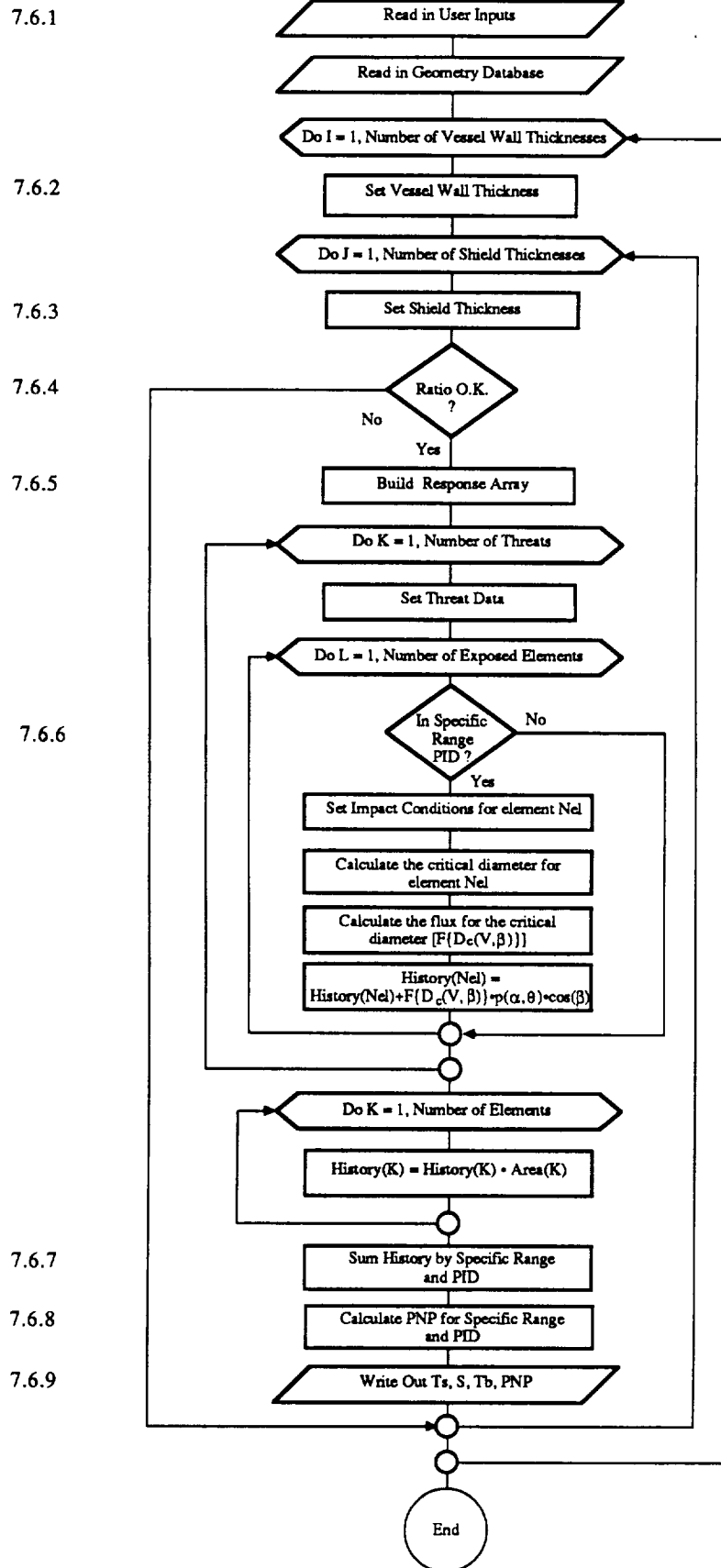


Figure 7.6-1. CONTOUR Flow Chart.

penetration functions. If the ratio is outside of the range, then this combination is skipped and the next one is calculated.

7.6.5 Build Response Array

The response array (RTABLE) is built using the RESPONSE code as a subroutine in the CONTOUR code. The only modification to the RESPONSE code is the elimination of the input and output subroutines.

7.6.6 Inside Range

If the element is not within the specified element ID range or if the element is within the range but has a property ID other than the one specified, then the element is skipped and the next element evaluated.

7.6.7 Sum History by Specific Range and PID

For the specific range of element IDs, the History array is summed up. This represents the total average number of penetrating particles per unit time for the specified range.

7.6.8 Calculate the PNP for the Specific Range and PID

The PNP for the range is calculated using the user specified exposure time.

7.6.9 Write Out Ts, S, Tb, PNP

For each shield and vessel wall combination (within the specified bounds stated in sec. 7.6.4), the shield thickness (Ts), shield standoff (S), vessel wall thickness (Tb), and associated PNP is written out to the screen and the summary file.

D180-30550-4

REFERENCES

- 1 "Final Report-Space Station Integrated Wall Design and Penetration Damage Control," Contract NAS8-36426, The Boeing Company, July 1987.

D180-30550-4

This page left intentionally blank.

D180-30550-4

APPENDIX A

SUPERTAB Universal File Format

Copyright 1983, Structural Dynamics Research Corporation

PRECEDING PAGE BLANK NOT FILMED

```

-1
 2
SDRC__I-DEAS 2.5B: M

ME 2.54000E-02
-1
-1
18
      6      0      0      15      1
REF
-1.00000E+02  0.00000E+00 -1.00000E+02 -1.00000E+02 -5.00000E+00 -1.0
-1.00000E+02 -5.00000E+00  0.00000E+00
-1
-1
15
      1      6      6      11  4.36746E+02  7.45440E+00  6.
      2      6      6      11  4.36746E+02  6.43840E+00  5.
      .      .      .      .      .      .
      .      .      .      .      .      .
      .      .      .      .      .      .
      2769      6      6      11  3.96919E+02  6.43840E+00  6.
      2770      6      6      11  3.96919E+02  6.91820E+00  6.
      2771      6      6      11  3.96919E+02  6.43840E+00  5.
-1
-1
71
      1      2      91      1      1      7      3
      266      47      48      .      .      .
      2      2      91      1      1      7      3
      52      266      267      .      .      .
      .      .      .      .      .      .
      .      .      .      .      .      .
      .      .      .      .      .      .
      5371      2      91      1      1      7      3
      2755      2770      2767      .      .      .
-1

```

Figure A-1. SUPERTAB Example – Universal File Format

```

-1
73
SUPERTAB 8.0 - MS - 0729830004
      0      1      13      8
0.100000E+01 0.000000E+00 0.000000E+00 0.000000E+00 0.100000E-34 0.10
0.100000E-34 0.000000E+00 0.000000E+00 0.000000E+00 0.100000E-34 0.10
0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.00
0.000000E+00
      0      5      13      8
0.100000E+01 0.000000E+00 0.000000E+00 0.000000E+00 0.100000E-34 0.10
0.100000E-34 0.000000E+00 0.000000E+00 0.000000E+00 0.100000E-34 0.10
0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00 0.00
0.000000E+00
      1      1      1      11
0.206800E+12 0.290000E+00 0.782000E+03 0.100000E-34 0.361000E-05 0.10
0.100000E-34 0.000000E+00 0.100000E-34 0.000000E+00 0.360000E+05
-1
-1
74
SUPERTAB 8.0 - MS - 0729830004
      1      91      1      1      1
      2      91      2      1      1
      .      .      .      .      .
      .      .      .      .      .
      .      .      .      .      .
      .      .      .      .      .
      5370      91      5370      1      1
      5371      91      5371      1      1
-1

```

Figure A-1. SUPERTAB Example – Universal File Format (Continued)

D180-30550-4

DATASET TYPE: 2
 DESCRIPTION: PROGRAM HEADER
 SUPERTAB, OUTPUT DISPLAY, ETC.
 RECORD 1: FORMAT (20A1)
 FIELD 1 — 20 CHARACTER PROGRAM VERSION IDENTIFIER.
 RECORD 2: FORMAT(80A1)
 FIELD 1 — 80 CHARACTER MODEL IDENTIFICATION.
 RECORD 3: FORMAT(2A1,E13.5)
 FIELD 1 — MODEL UNITS OF DATA
 IN = INCHES
 MM = MILLIMETERS
 ME = METERS
 UD = USER DEFINED SCALE
 FIELD 2 — MODEL UNITS CONVERSION FACTOR
 (MODEL UNITS PER INCH)

EXAMPLE:
 -1
 2
 SDRC/SUPERTAB X.Y.Z
 EXAMPLE OF PROGRAM HEADER DATASET
 IN 1.00000E+00
 -1

DATASET TYPE: 15
 DESCRIPTION: NODES
 RECORD 1: FORMAT(4I10,3E13.5)
 FIELD 1 — NODE TAG (NUMBER)
 FIELD 2 — DEFINITION COORDINATE SYSTEM
 FIELD 3 — DISPLACEMENT COORDINATE SYSTEM
 FIELD 4 — COLOR
 FIELDS 5-7 — 3-DIMENSIONAL COORDINATES OF NODE IN THE DEFINITION
 SYSTEM

RECORD 1 IS REPEATED FOR EACH NODE IN THE MODEL.

EXAMPLE:

-1							
15							
	1	0	0	8	0.00000E+00	0.00000E+00	0.00000E+00
	2	0	0	8	5.00000E-01	0.00000E+00	-5.00000E-02
	.						
	.						
	.						
	100	0	0	8	1.20000E+01	1.20000E+01	-4.50000E+00
-1							

D180-30550-4

DATASET TYPE: 18

DESCRIPTION: COORDINATE SYSTEMS

RECORD 1: FORMAT(5I10)

FIELD 1 — COORDINATE SYSTEM TAG (NUMBER)
 FIELD 2 — COORDINATE SYSTEM TYPE
 FIELD 3 — REFERENCE COORDINATE SYSTEM TAG
 FIELD 4 — COLOR
 FIELD 5 — METHOD OF DEFINITION

RECORD 2: FORMAT(20A1)

FIELD 1 — 20 CHARACTER COORDINATE SYSTEM IDENTIFICATION FOR
 METHOD OF DEFINITION = 1

RECORD 3: FORMAT(6E13.5)

TOTAL OF 9 COORDINATE SYSTEM DEFINITION PARAMETERS.

FIELDS 1-3 — ORIGIN OF NEW SYSTEM SPECIFIED IN REFERENCE SYSTEM
 FIELDS 4-6 — POINT ON +X AXIS OF THE NEW SYSTEM SPECIFIED IN
 REFERENCE SYSTEM
 FIELDS 7-9 — POINT ON +XZ PLANE OF THE NEW SYSTEM SPECIFIED IN
 REFERENCE SYSTEM

REPEAT RECORDS 1 THRU 3 FOR ALL COORDINATE SYSTEMS.

EXAMPLE:

```

      -1
      18
           3           1           0           8           1
SYS1
0.00000E+00 -1.00000E+00 7.00000E+00 -5.00000E+00 0.00000E+00 3.0000E+00
6.00000E+00 4.50000E+00 3.00000E+00
      -1
  
```

D180-30550-4

DATASET TYPE: 56
 DESCRIPTION: ANALYSIS DATA ON ELEMENTS
 RECORD 1: FORMAT (80A1)
 FIELD 1 — ID LINE 1
 RECORD 2: FORMAT (80A1)
 FIELD 1 — ID LINE 2
 RECORD 3: FORMAT (80A1)
 FIELD 1 — ID LINE 3
 RECORD 4: FORMAT (80A1)
 FIELD 1 — ID LINE 4
 RECORD 5: FORMAT (80A1)
 FIELD 1 — ID LINE 5
 RECORD 6: FORMAT (6I10)

DATA DEFINITION PARAMETERS

FIELD 1	—	MODEL TYPE 0: UNKNOWN 1: STRUCTURAL 2: HEAT TRANSFER 3: FLUID FLOW
FIELD 2	—	ANALYSIS TYPE 0: UNKNOWN 1: STATIC 2: NORMAL MODE 3: COMPLEX EIGENVALUE 4: TRANSIENT 5: FREQUENCY RESPONSE 6: BUCKLING
FIELD 3	—	DATA CHARACTERISTIC 0: UNKNOWN 1: SCALAR 2: 3 DOF GLOBAL TRANSLATION VECTOR 3: 6 DOF GLOBAL TRANSLATION & ROTATION VECTOR 4: SYMMETRIC GLOBAL TENSOR 5: GENERAL GLOBAL TENSOR
FIELD 4	—	SPECIFIC DATA TYPE 0: UNKNOWN 1: UNKNOWN 2: STRESS 3: STRAIN 4: ELEMENTAL FORCE 5: TEMPERATURE 6: HEAT FLUX 7: STRAIN ENERGY 8: DISPLACEMENT 9: REACTION FORCE 10: KINETIC ENERGY 11: VELOCITY 12: ACCELERATION
FIELD 5	—	DATA TYPE 2: REAL 5: COMPLEX
FIELD 6	—	NUMBER OF DATA VALUES FOR EACH POSITION ON THE ELEMENT (NDV)

RECORDS 7 AND 8 ARE ANALYSIS TYPE SPECIFIC

GENERAL FORM

D180-30550-4

RECORD 7: FORMAT (8I10)
 FIELD 1 — NUMBER OF INTEGER DATA VALUES
 1 < OR = NINT < X OR = 10
 FIELD 2 — NUMBER OF REAL DATA VALUES
 1 < OR = NRVAL < OR = 12
 FIELDS 3-N — TYPE SPECIFIC INTEGER PARAMETERS
 RECORD 8: FORMAT (6E13.5)
 FIELDS 1-N — TYPE SPECIFIC REAL PARAMETERS

FOR ANALYSIS TYPE = 0, UNKNOWN

RECORD 7:
 FIELD 1 — 1
 FIELD 2 — 1
 FIELD 3 — ID NUMBER
 RECORD 8:
 FIELD 1 — 0.0

FOR ANALYSIS TYPE = 1, STATIC

RECORD 7:
 FIELD 1 — 1
 FIELD 2 — 1
 FIELD 3 — LOAD CASE NUMBER
 RECORD 8:
 FIELD 4 — 0.0

FOR ANALYSIS TYPE = 2, NORMAL MODE

RECORD 7:
 FIELD 1 — 2
 FIELD 2 — 3
 FIELD 3 — LOAD CASE NUMBER
 FIELD 4 — MODE NUMBER
 RECORD 8:
 FIELD 1 — FREQUENCY (HERTZ)
 FIELD 2 — MODAL MASS
 FIELD 3 — MODAL DAMPING

FOR ANALYSIS TYPE = 3, COMPLEX EIGENVALUE

RECORD 7:
 FIELD 1 — 2
 FIELD 2 — 6
 FIELD 3 — LOAD CASE NUMBER
 FIELD 4 — MODE NUMBER
 RECORD 8:
 FIELD 1 — REAL PART EIGENVALUE
 FIELD 2 — IMAGINARY PART EIGENVALUE
 FIELD 3 — REAL PART OF MODAL A
 FIELD 4 — IMAGINARY PART OF MODAL A
 FIELD 5 — REAL PART OF MODAL B
 FIELD 6 — IMAGINARY PART OF MODAL B

FOR ANALYSIS TYPE = 4, TRANSIENT

RECORD 7:
 FIELD 1 — 2
 FIELD 2 — 1
 FIELD 3 — LOAD CASE NUMBER
 FIELD 4 — TIME STEP NUMBER
 RECORD 8:
 FIELD 1 — TIME (SECONDS)

FOR ANALYSIS TYPE = 5, FREQUENCY RESPONSE

RECORD 7:

FIELD 1 — 2
 FIELD 2 — 1
 FIELD 3 — LOAD CASE NUMBER
 FIELD 4 — FREQUENCY STEP NUMBER

RECORD 8:

FIELD 1 — FREQUENCY (HERTZ)

FOR ANALYSIS TYPE = 6, BUCKLING

RECORD 7:

FIELD 1 — 1
 FIELD 2 — 1
 FIELD 3 — LOAD CASE NUMBER

RECORD 8:

FIELD 1 — EIGENVALUE

RECORD 9:

FORMAT (2I10)
 FIELD 1 — ELEMENT NUMBER
 FIELD 2 — NUMBER OF DATA VALUES FOR THIS ELEMENT (NVAL)

RECORD 10:

FORMAT(6E13.5)
 FIELDS 1-N — DATA ON ELEMENT (NVAL REAL OR COMPLEX VALUES)

RECORDS 9 AND 10 ARE REPEATED FOR ALL ELEMENTS.

- NOTES:
1. D LINES MAY NOT BE BLANK. IF NO INFORMATION IS "NONE" MUST APPEAR IN COLUMNS 1-4.
 2. FOR COMPLEX DATA THERE WILL BE 2*NVAL DATA ITEMS. THE ORDER IS REAL PART FOR VALUE 1, IMAGINARY PART FOR VALUE 1, REAL PART FOR VALUE 2, IMAGINARY PART FOR VALUE 2, ETC.
 3. THE ORDER OF VALUES FOR VARIOUS DATA CHARACTERISTICS IS:
 3 DOF GLOBAL VECTOR: X, Y, Z
 6 DOF GLOBAL VECTOR: X, Y, Z, RX, RY, RZ
 SYMMETRIC GLOBAL TENSOR: SXX, SXY, SYX, SYZ, SYZ, SZZ
 GENERAL GLOBAL TENSOR: SXX, SYX, SZX, SXY, SYY, SZY, SXZ, SYZ, SZZ
 4. ID LINE 1 ALWAYS APPEARS ON PLOTS IN OUTPUT DISPLAY.
 5. IF SPECIFIC DATA TYPE IS "UNKNOWN," ID LINE 2 IS DISPLAYED AS DATA TYPE IN OUTPUT DISPLAY.
 6. TYPICAL FORTRAN I/O STATEMENTS FOR THE DATA SECTIONS ARE:

```

      READ (LUN, 1000) NUM, NVAL
      WRITE
    1000 FORMAT (2I10)
      READ (LUN, 1010) (VAL(I), I = 1, NVAL)
      WRITE
    1010 FORMAT (6E13.5)
    WHERE: NUM IS ELEMENT NUMBER
           NVAL IS NUMBER OF REAL OR COMPLEX DATA
           VALUES FOR THIS ELEMENT (MAX = 90)
           VAL IS REAL OR COMPLEX DATA ARRAY
    
```
 7. DATA CHARACTERISTIC VALUES IMPLY THE FOLLOWING VALUES OF NDV:
 SCALAR: 1
 3 DOF GLOBAL VECTOR: 3
 6 DOF GLOBAL VECTOR: 6
 SYMMETRIC GLOBAL TENSOR: 6
 GENERAL GLOBAL TENSOR: 9
 8. DATA ON 2D TYPE ELEMENTS MAY HAVE MULTIPLE VALUES THROUGH THE ELEMENT THICKNESS. IN THESE CASES, NVAL = NDVNPOS WHERE NPOS IS NUMBER OF POSITIONS THROUGH ELEMENT. NPOS IS ALWAYS 1 FOR SOLIDS. THE ORDER OF THE DATA IS NDV VALUES FOR POSITION 1, NDV VALUES FOR POSITION 2, ETC. THE ORDER OF THE NODES DEFINES AN OUTWARD NORMAL WHICH SPECIFIES THE ORDER FROM POSITION 1 TO NPOS.

9. ANY RECORD WITH ALL 0.0'S DATA ENTRIES NEED NOT (BUT MAY) APPEAR.
10. A DIRECT RESULT OF 9 IS THAT IF NO RECORD 9 ~~2~~ 10 APPEARS, ALL DATA FOR THE DATA SET IS 0.0.
11. WHEN NEW ANALYSIS TYPES ARE ADDED, RECORD 7 FIELDS 1 AND 2 ARE ALWAYS < OR = 1 WITH DUMMY INTERGER AND REAL ZERO DATA IF DATA IS NOT REQUIRED. IF COMPLEX DATA IS NEEDED, IT IS TREATED AS TWO REAL NUMBERS, REAL PART FOLLOWED BY IMAGINARY POINT.
12. DATALOADER USE THE FOLLOWING ID LINE CONVENTION:
 1. (80A1) MODEL IDENTIFICATION
 2. (80A1) RUN IDENTIFICATION
 3. (80A1) RUN DATE/TIME
 4. (80A1) LOAD CASE NAME

FOR STATIC:

5. (17H LOAD CASE NUMBER;, I10)

FOR NORMAL MODE:

5. (10H MODE SAME, I10, 10H FREQUENCY, E13.5)
13. MAXIMUM VALUE FOR NDV IS 9.
MAXIMUM VALUE FOR NVAL IS 90.
14. TYPICAL FORTRAN I/O STATEMENTS FOR PROCESSING RECORDS 7 AND 8.
 READ (LUN, 1000) NINT, NRVAL, (IPAR(I), I = 1, NINT)
 1000 FORMAT (8I10)
 READ (LUN, 1010) (NRVAL(I), I = 1, NRVAL)
 1010 FORMAT (6E13.5)

DATASET TYPE: 71

DESCRIPTION: CONNECTIVITY

RECORD 1: FORMAT(7I10)

FIELD 1 — ELEMENT TAG (NUMBER)
 FIELD 2 — CONNECTIVITY NUMBER
 FIELD 3 — TYPE REFERENCE (NUMBER)
 FIELD 4 — PROPERTY REFERENCE (NUMBER)
 FIELD 5 — MATERIAL REFERENCE (NUMBER)
 FIELD 6 — COLOR
 FIELD 7 — NUMBER OF NODES ON ELEMENT

RECORD 2: FORMAT (8I10)

FIELDS 1-N — NODE NUMBERS OF NODES DEFINING ELEMENT

THE ENTIRE SET OF RECORDS IS REPEATED FOR EACH ELEMENT OF THE MODEL.

EXAMPLE:

```

-1
71
    1      19      1      1      1      8      8
   11     12     13     16     21     20     19    15
    2      12      2      2      1      8     16
   31     32     33     34     35     36     37    38
   39     40     41     42     43     44     45    46
    .
    .
    .
  124     19      1      1      1      8      8
    9     10     11     15     19     18     17    14
-1

```

D180-30550-4

DATASET TYPE: 73
 DESCRIPTION: ELEMENT PROPERTY VALUE ENTRIES
 RECORD 1: FORMAT(40A2)
 FIELD 1 — PARENT FINITE ELEMENT SET SERIAL NUMBER
 RECORD 2: FORMAT (4I10)
 FIELD 1 — ENTRY TYPE
 0 — PHYSICAL
 1 — ISOTROPIC MATERIAL
 2 — ORTHOTROPIC MATERIAL
 3 — ANISOTROPIC MATERIAL
 FIELD 2 — ENTRY ID
 FIELD 3 — PHYSICAL OR MATERIAL DESCRIPTOR ID
 FIELD 4 — # OF PROPERTIES IN VALUE ENTRY
 RECORD 3: FORMAT (6E13.6)
 FIELD 1<n<6 — VALUES AS NEEDED
 RECORD 4: FORMAT(I10,35A2)
 FIELD 1 — SYMBOL COUNT
 FIRST RECORD OF VALUE — TOTAL SYMBOLS IN VALUE
 EACH SUCCESSIVE RECORD — TOTAL FOR THAT RECORD
 ONE RECORD TYPE 1 APPEARS PER DATASET. ONE RECORD TYPE 2
 APPEARS FOR EACH VALUE ENTRY IN THE DATASET. AS MANY
 RECORD TYPE 3 AND RECORD TYPE 4 APPEAR PER RECORD TYPE 2
 AS IS NECESSARY TO TRANSFER THE PROPERTY VALUES IN EACH
 ENTRY.

FINITE ELEMENT PHYSICAL PROPERTIES

PROPERTY DESCRIPTOR ELEMENT FAMILY AND INDEXES

1	NULL PROPERTY TABLE NUMBER OF PROPERTIES = 0	NUMBER OF VALUES = 0
2	LINEAR BEAM NUMBER OF PROPERTIES = 37 Property Indices 2,3,4,7,26,5,6,8,9,10,11,12,19,13,20,14,15,16,17, 18,51,58,94,59,95,60,96,25,27,83,84,85,86,87,91,92,93	NUMBER OF VALUES = 57
3	NODE TO NODE SPRING NUMBER OF PROPERTIES = 8 Property Indices 22,21,63,64,53,54,55,90	NUMBER OF VALUES = 15
4	SOLID NUMBER OF PROPERTIES = 3 Property Indices 49,88,89	NUMBER OF VALUES = 5
5	ROD NUMBER OF PROPERTIES = 4 Property Indices 2,7,24,25	NUMBER OF VALUES = 4
6	TAPERED BEAM NUMBER OF PROPERTIES = 65 Property Indices 28,30,32,97,99,38,34,65,107,67,109,68,111,69,113 70,115,75,40,79,77,101,103,105,44 10,11,12,19,13,20,14,15,16,17,36,37,42,43,81 29,31,33,98,100,39,35,66,108,71,110,72,112,73,114,74 116,76,41,80,78,102,104,106,45	NUMBER OF VALUES = 93
7	CURVED BEAM NUMBER OF PROPERTIES = 10 Property Indices 2,3,4,7,5,6,18,25,8,9	NUMBER OF VALUES = 11
8	PIPE NUMBER OF PROPERTIES = 5 Property Indices 191,192,193,194,25	NUMBER OF VALUES = 5
9	PHYSICAL PROPERTIES FOR DATA FORMATTER NUMBER OF PROPERTIES = 49 Property Indices 201,202,203,204,205,206,207,209,209,210,211,212,213 214,215,216,217,218,219,220,221,222,223,224,225,226 227,228,229,230,231,232,233,234,235,236,237,238,239 240,241,242,243,244,245,246,247,248,249	NUMBER OF VALUES = 48

10	PLANE STRESS NUMBER OF PROPERTIES = 2 Property Indices 1,49	NUMBER OF VALUES = 7
<hr/>		
PROPERTY DESCRIPTOR	ELEMENT FAMILY AND PROPERTY NUMBERS	
11	PLANE STRAIN NUMBER OF PROPERTIES = 2 Property Indices 1,49	NUMBER OF VALUES = 7
12	AXISYMMETRIC NUMBER OF PROPERTIES = 1 Property Indices 49	NUMBER OF VALUES = 3
13	THIN SHELL NUMBER OF PROPERTIES = 7 Property Indices 1,46,47,48,49,52,82	NUMBER OF VALUES = 11
14	FLAT PLATE NUMBER OF PROPERTIES = 3 Property Indices 1,46,49	NUMBER OF VALUES = 6
15	LUMPED MASS NUMBER OF PROPERTIES = 6 Property Indices 23,56,61,62,57,54	NUMBER OF VALUES = 13
16	THICK SHELL NUMBER OF PROPERTIES = 1 Property Indices 49	NUMBER OF VALUES = 3
17	NODE TO GROUND SPRING NUMBER OF PROPERTIES = 5 Property Indices 22,21,53,54,90	NUMBER OF VALUES = 12
18	NODE TO NODE DAMPER NUMBER OF PROPERTIES = 3 Property Indices 53,54,55	NUMBER OF VALUES = 3
19	NODE TO GROUND DAMPER NUMBER OF PROPERTIES = 2 Property Indices 53,54	NUMBER OF VALUES = 2

FINITE ELEMENT PHYSICAL PROPERTIES

INDEX	MENU PICK	DESCRIPTION
1	TK	THICKNESS (CAN INPUT UP TO 4 VALUES ANY 0.0 VALUE DEFAULTS TO PREVIOUS VALUE) DEFAULT VALUES 1.0,0.0,0.0,0.0
2	AR	AREA — CROSS SECTION AREA
3	IYY	MOMENT OF INERTIA ABOUT Y AXES
4	IZZ	MOMENT OF INERTIA ABOUT Z AXES
5	SRY	SHEAR AREA RATIO IN THE Y DIRECTION (SUPERB)
6	SRZ	SHEAR AREA RATIO IN THE Z DIRECTION (SUPERB)
7	TC	TORSIONAL CONSTANT
8	ECY	ECCENTRICITY Y — Y DISTANCE FROM SHEAR CENTER TO CENTROID (SUPERB)
9	ECZ	ECCENTRICITY Z — Z DISTANCE FROM SHEAR CENTER TO CENTROID (SUPERB)
10	OC	ORIENTATION CODE
11	OS	ORIENTATION SPECIFICATION
12	RFE	FORE END RELEASE FUNCTION (SUPERB,NASTRAN)
13	RAE	AFT END RELEASE FUNCTION (SUPERB,NASTRAN)
14	OFCF	OFFSET CODE FORE END (SUPERB)
15	OFCA	OFFSET CODE AFT END (SUPERB)
16	OFVF	OFFSET VECTOR FORE END
17	OFVA	OFFSET VECTOR AFT END
18	SRC	STRESS RECOVERY C (2 VALUES) (SUPERB,NASTRAN)

19	SFF	SPRING FUNCTION FORE END (6 VALUES) (SUPERB)
20	SFA	SPRING FUNCTION AFT END (6 VALUES) (SUPERB)
21	KIND	SPRING COEFFICIENT CODE
22	K	SPRING COEFFICIENT (8 VALUES)
23	M	MASS
24	CTS	COEFFICIENT FOR TORSIONAL STRESS
25	NSML	NONSTRUCTURAL MASS PER UNIT LENGTH (NASTRAN)
26	IXY	PRODUCT OF INERTIA (NASTRAN)
27	SAL	SURFACE AREA PER UNIT LENGTH (SUPERB)
28	ARF	AREA FORE END
29	ARA	AREA AFT END
30	IYYF	MOMENT OF INERTIA ABOUT Y AXES FORE END
31	IYYA	MOMENT OF INERTIA ABOUT Y AXES AFT END
32	IZZF	MOMENT OF INERTIA ABOUT Z AXES FORE END
33	IZZA	MOMENT OF INERTIA ABOUT Z AXES AFT END
34	TCF	TORSIONAL CONSTANT FORE END
35	TCA	TORSIONAL CONSTANT AFT END
36	SSF1	SHEAR STIFFNESS FACTOR XY PLANE
37	SSF2	SHEAR STIFFNESS FACTOR XZ PLANE
38	IXYF	PRODUCT OF INERTIA FORE END
39	IXYA	PRODUCT OF INERTIA AFT END
40	WCF	WARPING COEFFICIENT FORE END
41	WCA	WARPING COEFFICIENT AFT END
42	SRT1	SHEAR RELIEF DUE TO TAPER XY PLANE
43	SRT2	SHEAR RELIEF DUE TO TAPER XZ PLANE
44	NAF	COORDINATES OF NEUTRAL AXIS FORE END
45	NAA	COORDINATES OF NEUTRAL AXIS AFT END
46	BSP	BENDING STIFFNESS PARAMETER (NASTRAN)
47	TSMT	TRANSVERSE SHEAR THICKNESS DIVIDED BY MEMBRANE THICKNESS (NASTRAN)
48	NSMA	NONSTRUCTURAL MASS PER UNIT AREA (NASTRAN)
49	MOV	MATERIAL ORIENTATION VECTOR, REAL, 3 VALUES
51	ERTC	EFFECTIVE RADIUS IN TORSION C (SUPERB,SAGS)
52	Z12	FIBER DISTANCES FOR STRESS COMPUTATION, 2 VALUES (NASTRAN)
53	GE	DAMPING COEFFICIENT (NASTRAN)
54	C1	COMPONENT NUMBER 1 (NASTRAN)
55	C2	COMPONENT NUMBER 2 (NASTRAN)
56	MICS	MASS INERTIA COORDINATE SYSTEM (SUPERB)
57	MIM	MASS INERTIA MATRIX (SUPERB)
58	SRD	STRESS RECOVERY D (2 VALUES) (NASTRAN)
59	SRE	STRESS RECOVERY E (2 VALUES) (NASTRAN)
60	SRF	STRESS RECOVERY F (2 VALUES) (NASTRAN)
61	MOC	MASS OFFSET CODE (SUPERB)
62	MO	MASS OFFSET (3 VALUES) SUPERB
63	SOC	SPRING ORIENTATION CODE (SUPERB)
64	SOS	SPRING ORIENTATION SPECIFICATON (3 VALUES) SUPERB
65	NSMF	NONSTRUCTURAL MASS PER UNIT LENGTH FORE END (NASTRAN)
66	NSMA	NONSTRUCTURAL MASS PER UNIT LENGTH AFT END (NASTRAN)
67	SRCF	STRESS RECOVERY C FORE END (2 VALUES) (NASTRAN)
68	SRDF	STRESS RECOVERY D FORE END (2 VALUES) (NASTRAN)
69	SREF	STRESS RECOVERY E FORE END (2 VALUES) (NASTRAN)
70	SRFF	STRESS RECOVERY F FORE END (2 VALUES) (NASTRAN)
71	SRCA	STRESS RECOVERY C AFT END (2 VALUES) (NASTRAN)
72	SRDA	STRESS RECOVERY D AFT END (2 VALUES) (NASTRAN)

73	SREA	STRESS RECOVERY E AFT END (2 VALUES) (NASTRAN)
74	SRFA	STRESS RECOVERY F AFT END (2 VALUES) (NASTRAN)
75	NSIF	NONSTRUCTURAL MASS MOMENT OF INERTIA FORE END (NASTRAN)
76	NSIA	NONSTRUCTURAL MASS MOMENT OF INERTIA AFT END (NASTRAN)
77	NCGF	NONSTRUCTURAL MASS CENTER OF GRAVITY FORE END (NASTRAN)
78	NCGA	NONSTRUCTURAL MASS CENTER OF GRAVITY AFT END (NASTRAN)
79	WSAF	WARPING VARIABLE POINT FORE END (NASTRAN)
80	WSBA	WARPING VARIABLE POINT AFT END (NASTRAN)
81	SOO	STRESS OUTPUT OPTION (ALPHANUMERIC) (NASTRAN)
82	EFS	ELASTIC FOUNDATION STIFFNESS (ANSYS)
83	TKZ	BEAM THICKNESS Z DIRECTION (ANSYS)
84	THY	BEAM THICKNESS Y DIRECTION (ANSYS)
85	IS	INITIAL STRAIN (ANSYS)
86	SDCZ	SHEAR DEFLECTION CONSTANT Z DIRECTION (ANSYS)
87	SDCY	SHEAR DEFLECTION CONSTANT Y DIRECTION (ANSYS)
88	IN	INTEGRATION NETWORK (NASTRAN)
89	LOST	LOCATION FOR STRESS OUTPUT (NASTRAN)
90	SSC	SPRING STRESS COEFFICIENT (ANSYS)
91	WC	WARPING COEFFICIENT (SAGS)
92	FIX	DEGREE OF FIXITY (SAGS)
93	CSC	COMBINED STRESS CODE (SAGS)
94	ERTD	EFFECTIVE RADIUS IN TORSION D (SAGS)
95	ERTF	EFFECTIVE RADIUS IN TORSION E (SAGS)
96	ERTF	EFFECTIVE RADIUS IN TORSION F (SAGS)
97	SRFY	SHEAR AREA RATION Y FORE END (SAGS)
98	SRYA	SHEAR AREA RATION Y AFT END (SAGS)
99	SRZF	SHEAR AREA RATION Z FORE END (SAGS)
100	SRZA	SHEAR AREA RATION Z AFT END (SAGS)
101	FIXF	DEGREE OF FIXITY FORE END (SAGS)
102	FISA	DEGREE OF FIXITY AFT END (SAGS)
103	ECYF	ECCENTRICITY Y FORE END (SAGS)
104	ECYA	ECCENTRICITY Y AFT END (SAGS)
105	ECZF	ECCENTRICITY Z FORE END (SAGS)
106	ECZA	ECCENTRICITY Z AFT END (SAGS)
107	CSCF	COMBINED STRESS CODE FORE END (SAGS)
108	CSCA	COMBINED STRESS CODE AFT END (SAGS)
109	ERCF	EFFECTIVE RADIUS IN TORSION C FORE END (SAGS)
110	ERCA	EFFECTIVE RADIUS IN TORSION C AFT END (SAGS)
111	ERDF	EFFECTIVE RADIUS IN TORSION D FORE END (SAGS)
112	ERDA	EFFECTIVE RADIUS IN TORSION D AFT END (SAGS)
113	EREF	EFFECTIVE RADIUS IN TORSION E FORE END (SAGS)
114	EREA	EFFECTIVE RADIUS IN TORSION E AFT END (SAGS)
115	ERFF	EFFECTIVE RADIUS IN TORSION F FORE END (SAGS)
116	ERFA	EFFECTIVE RADIUS IN TORSION F AFT END (SAGS)
191	OD	OUTER DIAMETER — PIPE PROPERTY
192	WT	WALL THICKNESS — PIPE PROPERTY
193	SIFF	STRESS INTENSITY FACTOR FORE END — PIPE PROPERTY
194	SIFA	STRESS INTENSITY FACTOR AFT END — PIPE PROPERTY
201	P1	PHYSICAL PROPERTY 1 FOR DATA FORMATTER
202	P2	PHYSICAL PROPERTY 2 FOR DATA FORMATTER
203	P3	PHYSICAL PROPERTY 3 FOR DATA FORMATTER
204	P4	PHYSICAL PROPERTY 4 FOR DATA FORMATTER
205	P5	PHYSICAL PROPERTY 5 FOR DATA FORMATTER
206	P6	PHYSICAL PROPERTY 6 FOR DATA FORMATTER

207	P7	PHYSICAL PROPERTY 7 FOR DATA FORMATTER
208	P8	PHYSICAL PROPERTY 8 FOR DATA FORMATTER
209	P9	PHYSICAL PROPERTY 9 FOR DATA FORMATTER
210	P10	PHYSICAL PROPERTY 10 FOR DATA FORMATTER
211	P11	PHYSICAL PROPERTY 11 FOR DATA FORMATTER
212	P12	PHYSICAL PROPERTY 12 FOR DATA FORMATTER
213	P13	PHYSICAL PROPERTY 13 FOR DATA FORMATTER
214	P14	PHYSICAL PROPERTY 14 FOR DATA FORMATTER
215	P15	PHYSICAL PROPERTY 15 FOR DATA FORMATTER
216	P16	PHYSICAL PROPERTY 16 FOR DATA FORMATTER
217	P17	PHYSICAL PROPERTY 17 FOR DATA FORMATTER
218	P18	PHYSICAL PROPERTY 18 FOR DATA FORMATTER
219	P19	PHYSICAL PROPERTY 19 FOR DATA FORMATTER
220	P20	PHYSICAL PROPERTY 20 FOR DATA FORMATTER
221	P21	PHYSICAL PROPERTY 21 FOR DATA FORMATTER
222	P22	PHYSICAL PROPERTY 22 FOR DATA FORMATTER
223	P23	PHYSICAL PROPERTY 23 FOR DATA FORMATTER
224	P24	PHYSICAL PROPERTY 24 FOR DATA FORMATTER
225	P25	PHYSICAL PROPERTY 25 FOR DATA FORMATTER
226	P26	PHYSICAL PROPERTY 26 FOR DATA FORMATTER
227	P27	PHYSICAL PROPERTY 27 FOR DATA FORMATTER
228	P28	PHYSICAL PROPERTY 28 FOR DATA FORMATTER
229	P29	PHYSICAL PROPERTY 29 FOR DATA FORMATTER
230	P30	PHYSICAL PROPERTY 30 FOR DATA FORMATTER
231	P31	PHYSICAL PROPERTY 31 FOR DATA FORMATTER
232	P32	PHYSICAL PROPERTY 32 FOR DATA FORMATTER
232	P33	PHYSICAL PROPERTY 33 FOR DATA FORMATTER
234	P34	PHYSICAL PROPERTY 34 FOR DATA FORMATTER
235	P35	PHYSICAL PROPERTY 35 FOR DATA FORMATTER
236	P36	PHYSICAL PROPERTY 36 FOR DATA FORMATTER
237	P37	PHYSICAL PROPERTY 37 FOR DATA FORMATTER
238	P38	PHYSICAL PROPERTY 38 FOR DATA FORMATTER
239	P39	PHYSICAL PROPERTY 39 FOR DATA FORMATTER
240	P40	PHYSICAL PROPERTY 40 FOR DATA FORMATTER
241	P41	PHYSICAL PROPERTY 41 FOR DATA FORMATTER
242	P42	PHYSICAL PROPERTY 42 FOR DATA FORMATTER
243	P43	PHYSICAL PROPERTY 43 FOR DATA FORMATTER
244	P44	PHYSICAL PROPERTY 44 FOR DATA FORMATTER
245	P45	PHYSICAL PROPERTY 45 FOR DATA FORMATTER
246	P46	PHYSICAL PROPERTY 46 FOR DATA FORMATTER
247	P47	PHYSICAL PROPERTY 47 FOR DATA FORMATTER
248	P48	PHYSICAL PROPERTY 48 FOR DATA FORMATTER
249	P49	PHYSICAL PROPERTY 49 FOR DATA FORMATTER

FINITE ELEMENT MATERIAL PROPERTIES

PROPERTY DESCRIPTOR ELEMENT FAMILY AND INDEXES

1	ISOTROPIC	
	NUMBER OF PROPERTIES = 11	NUMBER OF VALUES = 11
	Property Indices 501,502,503,504,505,506,507,508,509,510,617	
2	ISOTROPIC (NULL)	
	NUMBER OF PROPERTIES = 0	NUMBER OF VALUES = 0
3	ORTHOTROPIC	
	NUMBER OF PROPERTIES = 19	NUMBER OF VALUES = 19
	Property Indices 503,507,508,601,602,603,604,605,606,608,609,610,611,612,613,614,615,616,617	

4	ANISOTROPIC NUMBER OF PROPERTIES = 12 Property Indices 503,507,508,509,701,702,703,704,705,706,708,709	NUMBER OF VALUES = 37
5	ORTHOTROPIC (NULL) NUMBER OF PROPERTIES = 0	NUMBER OF VALUES = 0
6	ANISOTROPIC (NULL) NUMBER OF PROPERTIES = 0	NUMBER OF VALUES = 0
7	ISOTROPIC (MATERIAL FOR DATA FORMATTER ELEMENTS) NUMBER OF PROPERTIES = 49 Property Indices 301,302,303,304,305,306,307,308,309,310,311,312,313 314,315,316,317,318,319,320,321,322,323,324,325,326 327,328,329,330,331,332,333,334,335,336,337,338,339 340,341,342,343,344,345,346,347,348,349	NUMBER OF VALUES = 49

FINITE ELEMENT MATERIAL PROPERTIES

INDEX	MENU PICK	DESCRIPTION
501	E	MODULUS OF ELASTICITY
502	NU	POISSON RATIO
503	DEN	MASS DENSITY
504	G	SHEAR MODULUS
505	A	COEFFICIENT OF THERMAL EXPANSION
506	K	THERMAL CONDUCTIVITY
507	TREF	THERMAL EXPANSION REFERENCE TEMPERATURE (NASTRAN)
508	GE	STRUCTURAL ELEMENT DAMPING COEFFICIENT (NASTRAN)
509	CP	THERMAL CAPACITY PER UNIT VOLUME (NASTRAN)
510	YS	YIELD STRESS (DEFAULT = 36000.) (SAGS)
601	EX	MODULUS OF ELASTICITY X DIRECTION
602	EY	MODULUS OF ELASTICITY Y DIRECTION
603	EZ	MODULUS OF ELASTICITY Z DIRECTION
604	NUXY	POISSON RATIO XY PLANE
605	NUYZ	POISSON RATIO YZ PLANE
606	NUXZ	POISSON RATIO XZ PLANE
608	GXY	SHEAR MODULUS XY PLANE
609	GYZ	SHEAR MODULUS YZ PLANE
610	GXZ	SHEAR MODULUS XZ PLANE
611	AX	COEFFICIENT OF THERMAL EXPANSION X DIRECTION
612	AY	COEFFICIENT OF THERMAL EXPANSION Y DIRECTION
613	AZ	COEFFICIENT OF THERMAL EXPANSION Z DIRECTION
614	KX	THERMAL CONDUCTIVITY X DIRECTION
615	KY	THERMAL CONDUCTIVITY Y DIRECTION
616	KZ	THERMAL CONDUCTIVITY Z DIRECTION
617	Q	HEAT GENERATION RATE (SUPERB)
701	RW1	ROW 1 MATERIAL PROPERTY MATRIX (6 VALUES)
702	RW2	ROW 2 MATERIAL PROPERTY MATRIX (5 VALUES)
703	RW3	ROW 3 MATERIAL PROPERTY MATRIX (4 VALUES)
704	RW4	ROW 4 MATERIAL PROPERTY MATRIX (3 VALUES)
705	RW5	ROW 5 MATERIAL PROPERTY MATRIX (2 VALUES)
706	RW6	ROW 6 MATERIAL PROPERTY MATRIX (1 VALUE)
708	TEV	THERMAL EXPANSION VECTOR (6 VALUES)
709	KKM	THERMAL CONDUCTIVITY MATRIX (6 VALUES)
301	M1	MATERIAL PROPERTY 1 FOR DATA FORMATTER
302	M2	MATERIAL PROPERTY 2 FOR DATA FORMATTER
303	M3	MATERIAL PROPERTY 3 FOR DATA FORMATTER
304	M4	MATERIAL PROPERTY 4 FOR DATA FORMATTER

305	M5	MATERIAL PROPERTY 5 FOR DATA FORMATTER
306	M6	MATERIAL PROPERTY 6 FOR DATA FORMATTER
307	M7	MATERIAL PROPERTY 7 FOR DATA FORMATTER
308	M8	MATERIAL PROPERTY 8 FOR DATA FORMATTER
309	M9	MATERIAL PROPERTY 9 FOR DATA FORMATTER
310	M10	MATERIAL PROPERTY 10 FOR DATA FORMATTER
311	M11	MATERIAL PROPERTY 11 FOR DATA FORMATTER
312	M12	MATERIAL PROPERTY 12 FOR DATA FORMATTER
313	M13	MATERIAL PROPERTY 13 FOR DATA FORMATTER
314	M14	MATERIAL PROPERTY 14 FOR DATA FORMATTER
315	M15	MATERIAL PROPERTY 15 FOR DATA FORMATTER
316	M16	MATERIAL PROPERTY 16 FOR DATA FORMATTER
317	M17	MATERIAL PROPERTY 17 FOR DATA FORMATTER
318	M18	MATERIAL PROPERTY 18 FOR DATA FORMATTER
319	M19	MATERIAL PROPERTY 19 FOR DATA FORMATTER
320	M20	MATERIAL PROPERTY 20 FOR DATA FORMATTER
321	M21	MATERIAL PROPERTY 21 FOR DATA FORMATTER
322	M22	MATERIAL PROPERTY 22 FOR DATA FORMATTER
323	M23	MATERIAL PROPERTY 23 FOR DATA FORMATTER
324	M24	MATERIAL PROPERTY 24 FOR DATA FORMATTER
325	M25	MATERIAL PROPERTY 25 FOR DATA FORMATTER
326	M26	MATERIAL PROPERTY 26 FOR DATA FORMATTER
327	M27	MATERIAL PROPERTY 27 FOR DATA FORMATTER
328	M28	MATERIAL PROPERTY 28 FOR DATA FORMATTER
329	M29	MATERIAL PROPERTY 29 FOR DATA FORMATTER
330	M30	MATERIAL PROPERTY 30 FOR DATA FORMATTER
331	M31	MATERIAL PROPERTY 31 FOR DATA FORMATTER
332	M32	MATERIAL PROPERTY 32 FOR DATA FORMATTER
333	M33	MATERIAL PROPERTY 33 FOR DATA FORMATTER
334	M34	MATERIAL PROPERTY 34 FOR DATA FORMATTER
335	M35	MATERIAL PROPERTY 35 FOR DATA FORMATTER
336	M36	MATERIAL PROPERTY 36 FOR DATA FORMATTER
337	M37	MATERIAL PROPERTY 37 FOR DATA FORMATTER
338	M38	MATERIAL PROPERTY 38 FOR DATA FORMATTER
339	M39	MATERIAL PROPERTY 39 FOR DATA FORMATTER
340	M40	MATERIAL PROPERTY 40 FOR DATA FORMATTER
341	M41	MATERIAL PROPERTY 41 FOR DATA FORMATTER
342	M42	MATERIAL PROPERTY 42 FOR DATA FORMATTER
343	M43	MATERIAL PROPERTY 43 FOR DATA FORMATTER
344	M44	MATERIAL PROPERTY 44 FOR DATA FORMATTER
345	M45	MATERIAL PROPERTY 45 FOR DATA FORMATTER
346	M46	MATERIAL PROPERTY 46 FOR DATA FORMATTER
347	M47	MATERIAL PROPERTY 47 FOR DATA FORMATTER
348	M48	MATERIAL PROPERTY 48 FOR DATA FORMATTER
349	M49	MATERIAL PROPERTY 49 FOR DATA FORMATTER

D180-30550-4

DATASET TYPE: 74
 DESCRIPTION: ELEMENT
 RECORD 1: FORMAT(40A2)
 FIELD 1 — PARENT FINITE ELEMENT SET SERIAL NUMBER
 RECORD 2-N: FORMAT(5I10)
 FIELD 1 — FINITE ELEMENT LABEL
 FIELD 2 — FINITE ELEMENT ANCHOR TYPE ID
 FIELD 3 — FINITE ELEMENT TOPOLOGY LABEL
 (IF ZERO, THEN ASSUMED TO BE SAME AS
 FINITE ELEMENT LABEL)
 FIELD 4 — FINITE ELEMENT PHYSICAL VALUE ID
 FIELD 5 — FINITE ELEMENT MATERIAL VALUE ID

FINITE ELEMENTS

ELEMENT DESCRIPTION*	ANCHOR TYPE	CONNECTIVITY #	PROPERTY DESCRIPTOR TYPES			
			PHYSICAL	ISO	ORTHO	ANISO
PSS,LT	42	2	10	1	3	4
PSS,PT	42	3	10	1	3	4
PSS,CT	43	3	10	1	3	0
PSS,LQ	44	5	10	1	3	4
PSS,PQ	45	6	10	1	3	4
PSS,CQ	40	7	10	1	3	0
PST,LT	51	2	11	1	3	0
PST,PT	52	3	11	1	3	0
PST,CT	53	4	11	1	3	0
PST,LQ	54	5	11	1	3	0
PST,PQ	55	6	11	1	3	0
PST,CQ	56	7	11	1	3	0
PLT,LT	61	2	14	1	3	4
PLT,PT	62	3	14	1	3	4
PLT,CT	63	1	14	1	3	4
PLT,LQ	64	5	14	1	3	4
PLT,PQ	65	6	14	1	3	4
PLT,CQ	66	7	14	1	3	0
MEM,LQ	71	5	13	1	3	0
AXI,LT	31	25	12	1	3	0
AXI,PT	82	26	12	1	3	0
AXI,LQ	84	27	12	1	3	0
AXI,PQ	85	28	12	1	3	0
TN,LT	91	2	13	1	3	4
TN,PT	92	3	13	1	3	4
TN,CT	93	4	13	1	3	4
TN,LQ	94	5	13	1	3	4
TN,PQ	95	6	13	1	3	4
TN,CQ	96	7	13	1	3	0
TK,LW	101	8	16	1	3	0
TK,PW	102	9	16	1	3	0
TK,CW	103	10	16	1	3	0
TK,LB	104	11	16	1	3	0
TK,PB	105	12	16	1	3	0
TK,CB	106	13	16	1	3	0
SOL,LT	111	14	4	1	3	4
SOL,LW	112	16	4	1	3	4
SOL,PW	113	17	4	1	3	4
SOL,CW	114	18	4	1	3	0

D180-30550-4

SOL,LB	115	19	4	1	3	4
SOL,PB	116	20	4	1	3	4
SOL,CB	117	21	4	1	3	4
RB	121	34	1	1	0	0
SP,NNS	131	29	3	2	5	6
SP,NGS	132	30	17	2	5	6
SP,NND	133	31	18	2	5	6
SP,NGD	134	32	19	2	5	6
SP,LM	135	33	15	2	5	6
PI,SI	31	1	8	1	0	0
PI,EL	32	1	8	1	0	0
BM,LB	21	1	2	1	0	0
BM,TB	22	1	6	1	0	0
BM,CB	23	1	7	1	0	0
RD	11	1	5	1	0	0

D180-30550-4

This page left intentionally blank.

D180-30550-4

APPENDIX B

Source Code for GEOMETRY Module

D180-30550-4

```

0001  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0002  C                                                              C
0003  C          GEOMETRY VER 2.51    06-30-1986          C
0004  C                                                              C
0005  C          BOEING AEROSPACE CO.          C
0006  C                                                              C
0007  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008  C
0009  C
0010  C  Geometry is a code for producing the geometry tables used by the
0011  C  BUMPER code for evaluating space structures impacted by man made
0012  C  orbital debris or meteoroids. The geometry of the structure is
0013  C  defined by a Supertab universal file which the code reads in. The
0014  C  coordinates of the grid points in the universal file must be in
0015  C  the basic coordinate sytem. The code outputs the threat information,
0016  C  the element id and pid, and each element's surface area. In
0017  C  addition for each threat considered the code outputs a list of the
0018  C  exposed elements and the cosine of the impact angle on each exposed
0019  C  element.
0020  C
0021  C  The code was developed under the NASA contract 'Space Station
0022  C  Integrated Wall Design Guide and Penetration Damage Control'
0023  C  by M.A. Wright and A.R. Coronado.
0024  C
0025  C
0026  C
0027  C  Common Block
0028  C
0029  C  Variable List
0030  C
0031  C    it = current relative threat angle case
0032  C    itype = analysis type , 1-man-made debris, 2-meteoroids
0033  C    nelm = total number of elements
0034  C    npe = number of potentially exposed elements for the current
0035  C          relative threat angle case
0036  C    nt = total number of relative threat angle cases
0037  C
0038  C
0039  C  Array list
0040  C
0041  C    element = global array containing the primary element data as read
0042  C              from Supertab universal file
0043  C
0044  C          1- x coordinate of grid 1
0045  C          2- y      "      "      "      "
0046  C          3- z      "      "      "      "
0047  C          4- x      "      "      "      2
0048  C          5- y      "      "      "      "
0049  C          6- z      "      "      "      "
0050  C          7- x      "      "      "      3
0051  C          8- y      "      "      "      "
0052  C          9- z      "      "      "      "
0053  C          10- a value of the unit normal vector
0054  C          11- b      "      "      "      "      "      "
0055  C          12- c      "      "      "      "      "      "
0056  C          13- x coordinate of the centroid
0057  C          14- y      "      "      "      "
0058  C          15- z      "      "      "      "
0059  C          16- square of the radius enclosing element, origin at cent.
0060  C          17- surface area
0061  C
0062  C    id = global array containing the element id, property id , and grid
0063  C          point id's

```

D180-30550-4

```

0064 C
0065 C      1- eid
0066 C      2- pid
0067 C      3- idg1
0068 C      4- idg2
0069 C      5- idg3
0070 C
0071 C      iel = global list of the elements containing exposure logical values
0072 C            for the current threat angle
0073 C
0074 C            1- frontside element
0075 C            2- exposed element
0076 C
0077 C      iexp = global list of the potentially exposed element pointers for
0078 C            the current threat angle sorted on the x and y transformed
0079 C            axis
0080 C
0081 C            1- sorted on x tranformed
0082 C            2- "      " y      "
0083 C
0084 C      point = global array of pointers , indicates where in IEXP an element
0085 C            is located
0086 C
0087 C            1- location in IEXP(1,I)
0088 C            2- location in IEXP(2,I)
0089 C
0090 C      transform = global array containing the element data in the coordinate
0091 C            system parallel to the current threat angle
0092 C
0093 C            1- cosine of the impact angle, measured from the normal
0094 C            2- x transformed coordinate for the centroid
0095 C            3- y      "      "      "      "
0096 C            4- z      "      "      "      "
0097 C            5- x      "      "      "      grid 1
0098 C            6- y      "      "      "      "
0099 C            7- z      "      "      "      "
0100 C            8- x      "      "      "      2
0101 C            9- y      "      "      "      "
0102 C           10- z      "      "      "      "
0103 C           11- x      "      "      "      3
0104 C           12- y      "      "      "      "
0105 C           13- z      "      "      "      "
0106 C
0107 C      threat = global array containing the threat data
0108 C
0109 C            1- theta angle, measured from space station velocity
0110 C               vector in the horizontal plane, radians
0111 C            2- phi angle, measured from the Earth normal, radians
0112 C            3- relative velocity of the threat particle with respect
0113 C               to the Space Station, km/sec
0114 C            4- probability of the relative threat occuring
0115 C
0116 C
0117 C      Main Program
0118 C
0119 C      Variable List
0120 C
0121 C      ifile = filename of the Supertab universal file
0122 C      ivar = variable in transform to sort on
0123 C      ofile = output filename
0124 C
0125 C
0126 C

```

D180-30550-4

```

0127 C
0128 INCLUDE 'COMMON1.BLK'
0147 C
0148 CHARACTER*80 IFILE,OFILE
0149 C
0150 C
0151 C Write header, obtain analysis type and input and output filenames
0152 C
0153 CALL HEADER ( IFILE,OFILE )
0154 C
0155 C Open files
0156 C
0157 10 OPEN (UNIT=2,FILE=IFILE,STATUS='OLD',ERR=20)
0158 C
0159 GO TO 50
0160 C
0161 C Error control on open
0162 C
0163 20 WRITE ( 6,30 )IFILE
0164 30 FORMAT ( /1X,'UNABLE TO OPEN SUPERTAB UNIVERSAL FILE ',A,
0165 1 /1X,'SUPERTAB UNIVERSAL FILENAME >' )
0166 READ ( 5,40 ) IFILE
0167 40 FORMAT (A)
0168 GO TO 10
0169 C
0170 50 OPEN (UNIT=4,FILE=OFILE,STATUS='UNKNOWN',FORM='UNFORMATTED',
0171 1 ERR=60)
0172 C
0173 REWIND 4
0174 C
0175 GO TO 75
0176 C
0177 C Error control on open
0178 C
0179 60 WRITE ( 6,70 )OFILE
0180 70 FORMAT ( /1X,'UNABLE TO OPEN OUTPUT FILE ',A,
0181 1 1X,'OUTPUT FILENAME >')
0182 READ ( 5,40 ) OFILE
0183 GO TO 50
0184 C
0185 C Create threat array
0186 C
0187 75 IF ( ITYPE.EQ.1 ) THEN
0188 CALL DTHREAT
0189 ELSE
0190 CALL MTHREAT
0191 END IF
0192 C
0193 C Read in Supertab data base
0194 C
0195 CALL DATA
0196 C
0197 C Calculate unit vector normals for all elements
0198 C
0199 CALL NORMAL
0200 C
0201 C Calculate centroids for all elements
0202 C
0203 CALL CENTROID
0204 C
0205 C Calculate the area of each element in global coordinates
0206 C
0207 CALL AREA

```

D180-30550-4

```

0208      C
0209      C Calculate maximum radius enclosing each element
0210      C
0211          CALL RADIUS
0212      C
0213      C This section determines hidden/exposed surfaces (elements)
0214      C
0215      C Initialize NPE to 0
0216      C
0217          NPE=0
0218      C
0219          DO 100 I=1,NT
0220      C
0221              IT=I
0222      C
0223      C Eliminate all elements whose normal vector points away from the
0224      C threat ( backside elements )
0225      C
0226          CALL BACKSIDE
0227      C
0228      C Transform remaining elements to a coordinate system parallel to the
0229      C threat
0230      C
0231          CALL TRANS
0232      C
0233      C Sort the potentially exposed elements on their transformed x distance
0234      C from the threat in descending order
0235      C
0236          IVAR=2
0237          CALL QSORT ( IVAR )
0238      C
0239      C Sort the potentially exposed elements on their transformed y distance
0240      C from the threat in descending order
0241      C
0242          IVAR=3
0243          CALL QSORT ( IVAR )
0244      C
0245      C Generate the element pointers for the sorted lists
0246      C
0247          DO 80 J=1,NPE
0248              POINT(1,(IEXP(1,J)))=J
0249              POINT(2,(IEXP(2,J)))=J
0250      80      CONTINUE
0251      C
0252      C Eliminate the elements that are in the shadow of other elements
0253      C
0254          CALL SHADOW
0255      C
0256      C Write data to the output file
0257      C
0258          CALL OUTPUT
0259      C
0260      C Next threat direction
0261      C
0262      C Write completed number to the screen
0263      C
0264          WRITE ( 6,90 ) IT
0265      90      FORMAT(1X,'THREAT CASE ',I4,' COMPLETED')
0266      C
0267      100      CONTINUE
0268      C
0269      C Write out location of output file to the screen
0270      C

```

D180-30550-4

```
0271      WRITE (6,110) OFILE
0272      110 FORMAT (/1X,'OUTPUT LOCATED IN FILE ', A)
0273      C
0274      C Close files and save
0275      C
0276      CLOSE ( UNIT = 2 , STATUS = 'KEEP' )
0277      CLOSE ( UNIT = 4 , STATUS = 'KEEP' )
0278      C
0279      END
```

D180-30550-4

```

0001      C
0002      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003      C
0004      SUBROUTINE HEADER (IFILE,OFIL)
0005      C
0006      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007      C
0008      C
0009      C Header writes the program header to the screen and reads in the
0010      C supertab universal file name and the output filename. It also
0011      C determines the analysis type.
0012      C
0013      C
0014      C Variable list
0015      C
0016      C answer = character string representing user input
0017      C ifile = character string for input file name
0018      C ofile = character string for output file name
0019      C
0020      C
0021      C
0022      INCLUDE 'COMMON1.BLK'
0041      C
0042      CHARACTER*80 ANSWER,IFILE,OFIL
0043      C
0044      C Write header to screen
0045      C
0046      WRITE ( 6,10 )
0047      10 FORMAT (/1X,'GEOMETRY VER 2.51')
0048      C
0049      C Determine analysis type, set default to 1 (debris)
0050      C
0051      15 WRITE ( 6,20 )
0052      20 FORMAT (/1X,'ANALYSIS TYPE ?',/2X,'1-DEBRIS <CR> ',/2X,
0053      1 '2-METEORIDS',/1X,'ANSWER 1 OR 2 >',$)
0054      C
0055      READ ( 5,30 ) ANSWER
0056      30 FORMAT (A)
0057      C
0058      IF ( ANSWER(1:1).EQ.' ' ) THEN
0059      ITYPE=1
0060      ELSE
0061      READ ( ANSWER(1:80),40 ) ITYPE
0062      40 FORMAT ( BN,I4 )
0063      END IF
0064      C
0065      C Check that input was correct
0066      C
0067      IF ( ITYPE.EQ.1 .OR. ITYPE.EQ.2 ) THEN
0068      CONTINUE
0069      ELSE
0070      WRITE ( 6,50 )
0071      50 FORMAT ( /1X,'INCORRECT INPUT' )
0072      GO TO 15
0073      END IF
0074      C
0075      C Read Supertab universal filename, or set to default to station.uni
0076      C
0077      WRITE ( 6,60 )
0078      60 FORMAT (/1X,'SUPERTAB UNIVERSAL FILENAME (CR=STATION.UNI) >',$)
0079      READ ( 5,30 ) IFILE
0080      IF ( IFILE(1:1).EQ.' ' ) IFILE='STATION.UNI'
0081      C

```

D180-30550-4

```
0082 C Read output file name, or set to default to station.gem
0083 C
0084 WRITE ( 6,70 )
0085 70 FORMAT (/1X,'OUTPUT FILENAME (CR=STATION.GEM) >',$)
0086 READ ( 5 , 30 )OFILE
0087 IF (OFILE(1:1).EQ.' ')OFILE='STATION.GEM'
0088 C
0089 C Finished
0090 C
0091 RETURN
0092 C
0093 END
```

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE DTHREAT
0005 C
0006 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007 C
0008 C
0009 C Dthreat builds the Threat array for man-made orbital debris
0010 C analysis. The code models only relative threats between -90 and 90
0011 C degrees in the horizontal plane measured from the space station
0012 C velocity vector. Orbital mechanics show that only these relative
0013 C threats can occur.
0014 C
0015 C The code calculates the velocity of the debris particle relative to
0016 C the space station. The probability of the relative threat occurring
0017 C is then derived from the NASA JSC-20001 velocity distribution. The
0018 C orbital altitude is assumed to be 500 km.
0019 C
0020 C The code requires that the relative velocity distribution be defined
0021 C in the user file DEB.VEL . The file is a list of third order curves
0022 C defining the various ranges of the distribution. There can be up
0023 C to 20 ranges. The data is read in free field format. The ranges must
0024 C be in ascending order. The record format is as follows :
0025 C
0026 C record 1-20 - nr,rmin,rmax,c1,c2,c3,c4
0027 C
0028 C where
0029 C
0030 C c1-c4 = third order curve constants
0031 C p(vr)= c1 + c2*vr + c3*vr**2 + c4*vr**3
0032 C nr = range number
0033 C rmax = max relative velocity in range nr
0034 C rmin = min relative velocity in range nr
0035 C
0036 C
0037 C
0038 C note: for variables in the common block see the main listing
0039 C
0040 C Variable List
0041 C
0042 C answer = character string representing user input
0043 C cp1 = cumulative relative threat angle probability at angle T1
0044 C cp2 = " " " " " " " " T2
0045 C c1,c2,c3,c4 = third order curve fit constants for the relative
0046 C velocity distribution at range nr
0047 C ic = counter
0048 C ip1 = pointer for angle T1 in the DATA array
0049 C ip2 = " " " T2 " " " "
0050 C ip12 = pointer for the next angle after ip1
0051 C ip22 = " " " " " " ip2
0052 C ncal = number of calculations for the DATA arrays
0053 C nr = current range number for the relative velocity distribution
0054 C pra = probability densisty of the current relative threat
0055 C prv = " " " " " " velocity
0056 C prob = probability of the current threat (+/- tinc/2) occurring
0057 C phi = relative threat angle measured from the Earth normal,radians
0058 C rat1,rat2 = intermediate linear interpolation variables
0059 C sum = sum of the current list program is evaluating
0060 C tc = theta increment used in the DATA array calculations, radians
0061 C thc = current theta , used in DATA array calculations, radians
0062 C theta = relative threat angle measured from the space station
0063 C velocity vector in the horizontal plane, radians

```

```

0064 C   tinc = theta increment in the THREAT array, radians
0065 C   t1 = theta plus tinc/2 ,radians
0066 C   t2 = " minus tinc/2, radians
0067 C   vmax = maximum relative velocity in VDIST, km/sec
0068 C   vmin = minimum " " " " " "
0069 C   vr = relative velocity of the debris particle with respect to the
0070 C   space station, km/sec
0071 C   vs = space station orbital velocity, km/sec
0072 C   v11,v12,v21,v22 = intermediate linear interpolation variables
0073 C   xmax = maximum relative velocity in range nr, km/sec
0074 C   xmin = minimum " " " " " "
0075 C
0076 C
0077 C   Array List
0078 C
0079 C   data = working list of the relative threat angle probability
0080 C   density and the cumulative relative angle probability.
0081 C   Used to calculate the Threat array.
0082 C   threat = global array containing the threat information, theta,phi,
0083 C   vr,prob
0084 C   vdist = array containing third order curve constants describing the
0085 C   the NASA JSC 20001 relative velocity distribution
0086 C
0087 C
0088 C
0089 C
0090 C   INCLUDE 'COMMON1.BLK'
0109 C
0110 C   DIMENSION VDIST(6,20),DATA(2,2000)
0111 C
0112 C   CHARACTER*90 ANSWER
0113 C
0114 C   PARAMETER (PI=3.1415926536)
0115 C
0116 C   Set the space station velocity
0117 C
0118 C   VS=7.50D0
0119 C
0120 C   Set Phi equal to a constant ( PI/2 )
0121 C
0122 C   PHI=PI/2.0
0123 C
0124 C   Set up the working array variables
0125 C
0126 C   NCAL=1000
0127 C   TC=PI/2.0/NCAL
0128 C
0129 C   Read in the number of uniform threats, set default to 45
0130 C
0131 C   40 WRITE ( 6,50 )
0132 C   50 FORMAT ( /1X,'NUMBER OF UNIFORM DEBRIS THREATS (CR=45) >','$ )
0133 C   READ ( 5,60 ) ANSWER
0134 C   60 FORMAT (A)
0135 C
0136 C   IF ( ANSWER(1:1).EQ.' ' ) THEN
0137 C     NT=45
0138 C   ELSE
0139 C     READ ( ANSWER(1:90),70 ) NT
0140 C   70 FORMAT ( BN,I6 )
0141 C   END IF
0142 C
0143 C   Check that the number of threats is less than 200
0144 C

```

```

0145         IF ( NT.GT.200 ) THEN
0146             WRITE ( 6,80 ) NT
0147     80     FORMAT ( /1X,'NUMBER OF THREATS (' ,I4,' ) IS OUTSIDE OF RANGE')
0148             GO TO 40
0149         END IF
0150     C
0151     C Calculate the Theta increment in the threat array
0152     C
0153         TINC=PI/NT
0154     C
0155     C Open the DEB.VEL file and read in the relative velocity distribution
0156     C data
0157     C
0158         OPEN ( UNIT=7,FILE='DEB.VEL',STATUS='OLD',ERR=100 )
0159     C
0160         GO TO 200
0161     C
0162     C Error control for open
0163     C
0164     100 WRITE ( 6,110 )
0165     110 FORMAT ( /1X,'DEBRIS VELOCITY DISTRIBUTION FILE DEB.VEL WAS',
0166         1      ' NOT FOUND'/' FILENAME ? >')
0167         READ ( 5,60 ) ANSWER
0168     C
0169         IF ( ANSWER(1:2).EQ.' ' ) GO TO 100
0170     C
0171         OPEN ( UNIT=7,FILE=ANSWER,STATUS='OLD',ERR=100 )
0172     C
0173     C Read the data, counting the number of ranges read
0174     C
0175     200 IC=0
0176         DO 225 I=1,20
0177             READ ( 7,*,END=250 )NR,(VDIST(J,I),J=1,6)
0178             IC=IC+1
0179     225 CONTINUE
0180     C
0181     250 CLOSE ( UNIT=7,STATUS='KEEP' )
0182     C
0183     C Determine the minimum and maximum allowable relative velocities
0184     C
0185         VMIN=VDIST(1,1)
0186         VMAX=VDIST(2,IC)
0187     C
0188     C Intialize the range pointer to 1
0189     C
0190         NR=1
0191     C
0192     C Calculate the relative threat angle probability density for the
0193     C working array
0194     C
0195         DO 400 I=1,NCAL+1
0196     C
0197     C Set the Theta angle and calculate the relative velocity. From
0198     C orbital mechanics it is known that the velocity of the station is
0199     C approximately equal to the velocity of the debris particle. Vr is
0200     C related to vs through vector addition.
0201     C
0202         THC=(I-1)*TC
0203         VR=2.0*VS*ABS(COS(THC))
0204     C
0205     C Check that vr is in the range of the relative velocity distribution
0206     C data
0207     C

```

```

0208             IF ( VR.LT.VMIN .OR. VR.GT.VMAX ) THEN
0209                 WRITE ( 6,275 )VR
0210             275             FORMAT ( /1X,'IMPACT VELOCITY OUTSIDE OF ALL RANGES ',/
0211                 1             1X,'VR=',E12.5)
0212                 STOP
0213             END IF
0214         C
0215         C   Get the current relative velocity range
0216         C
0217             300             XRMIN=VDIST(1,NR)
0218                 XRMAX=VDIST(2,NR)
0219         C
0220         C   Check that this is the correct range, if not increment accordingly
0221         C   being sure to not step outside of VDIST array
0222         C
0223                 IF (VR.LT.XRMIN) THEN
0224                     NR=NR-1
0225         C
0226                 IF ( NR.LT.1 ) THEN
0227                     WRITE ( 6,350 )
0228             350             FORMAT ( /1X,'CANNOT FIND CORRECT VELOCITY RANGE ' )
0229                     STOP
0230                 END IF
0231         C
0232                 GO TO 300
0233             END IF
0234         C
0235             IF (VR.GT.XRMAX) THEN
0236                 NR=NR+1
0237         C
0238                 IF ( NR.GT.IC ) THEN
0239                     WRITE ( 6,350 )
0240                     STOP
0241                 END IF
0242         C
0243                 GO TO 300
0244             END IF
0245         C
0246         C   Get curve fit constants
0247         C
0248                 C1=VDIST(3,NR)
0249                 C2=VDIST(4,NR)
0250                 C3=VDIST(5,NR)
0251                 C4=VDIST(6,NR)
0252         C
0253         C   Calculate the relative velocity probability density
0254         C
0255                 PRV=C1+C2*VR+C3*VR**2.0+C4*VR**3.0
0256         C
0257         C   Calculate the associated relative threat angle probability density,
0258         C   it is the relative velocity probability density multiplied by the
0259         C   absolute value of the derivative of the equation relating the two.
0260         C
0261                 PRA=PRV*2.0*VS*ABS(SIN(THC))
0262         C
0263         C   Store in the DATA array
0264         C
0265                 DATA(1,I)=PRA
0266         C
0267         C   Next threat
0268         C
0269             400 CONTINUE
0270         C

```

D180-30550-4

```

0271 C Use a trapezoid approximation to determine the area under the
0272 C probability density curve. The running sum of this value is the
0273 C cumulative probability. Store the cumulative probability in the
0274 C DATA array.
0275 C
0276 DATA(2,1)=0.0D0
0277 C
0278 SUM=0.0D0
0279 DO 500 I=1,NCAL
0280 PROB=0.50*TC*(DATA(1,I)+DATA(1,(I+1)))
0281 SUM=SUM+PROB
0282 DATA(2,(I+1))=SUM
0283 500 CONTINUE
0284 C
0285 C Use the DATA array to determine the THREAT array
0286 C
0287 DO 700 I=1,NT
0288 C
0289 C Set Theta and calculate the relative velocity
0290 C
0291 THETA=(TINC/2.0+(I-1)*TINC)-PI/2.0
0292 VR=2.0*VS*ABS(COS(THETA))
0293 C
0294 C Determine the location of the nearest angle in the data array
0295 C to the theta that is still less than theta
0296 C
0297 IP=ABS(THETA/TC)+1
0298 C
0299 C Set the threat angle increment , T1 to T2
0300 C
0301 T1=THETA-TINC/2.0
0302 T2=THETA+TINC/2.0
0303 C
0304 C Determine the location of nearest angle to T1 & T2 in the Data
0305 C array that is still less than T1 & T2
0306 C
0307 IP1=ABS(T1/TC)+1
0308 IP2=ABS(T2/TC)+1
0309 C
0310 C Determine the location of the next largest angle in the Data array
0311 C for T1 and T2. Checking that it is not outside the array.
0312 C
0313 IP12=IP1+1
0314 IF ( IP12.GT.NCAL+1 ) IP12=NCAL+1
0315 IP22=IP2+1
0316 IF ( IP22.GT.NCAL+1 ) IP22=NCAL+1
0317 C
0318 C Estimate the probability of the threat occurring by using the cumulative
0319 C probability values in the Data array and linear interpolation.
0320 C
0321 C Get the cumulative probability data from the Data array
0322 C
0323 V11=DATA(2,IP1)
0324 V12=DATA(2,IP12)
0325 V21=DATA(2,IP2)
0326 V22=DATA(2,IP22)
0327 C
0328 C Perform the interpolation
0329 C
0330 RAT1=ABS(T1/TC-INT(T1/TC))
0331 RAT2=ABS(T2/TC-INT(T2/TC))
0332 C
0333 CP1=RAT1*(V12-V11)+V11

```

```

0334      CP2=RAT2*(V22-V21)+V21
0335      C
0336      C   For angle increments spanning the 0.0 angle add the cumulative values.
0337      C   Else subtract the values.
0338      C
0339      IF ( IP.LE.1 ) THEN
0340          PROB=ABS(CP1+CP2)
0341      ELSE
0342          PROB=ABS(CP2-CP1)
0343      END IF
0344      C
0345      C   Store the data in the Threat array
0346      C
0347          THREAT(1,I)=THETA
0348          THREAT(2,I)=PHI
0349          THREAT(3,I)=VR
0350          THREAT(4,I)=PROB
0351      C
0352      C   Next threat
0353      C
0354      700 CONTINUE
0355      C
0356      C   Normalize the probabilitly data in the threat array to 1.0
0357      C
0358          SUM=0.0D0
0359          DO 800 I=1,NT
0360              SUM=SUM+THREAT(4,I)
0361      800 CONTINUE
0362      C
0363          DO 850 I=1,NT
0364              THREAT(4,I)=THREAT(4,I)/SUM
0365      850 CONTINUE
0366      C
0367      C   Finished
0368      C
0369          RETURN
0370      C
0371      END

```

```

0001      C                               D180-30550-4
0002      C
0003      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0004      C
0005      SUBROUTINE MTHREAT
0006      C
0007      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008      C
0009      C
0010      C  Mthreat builds the Threat array for meteoroid analysis. The code
0011      C  models the relative threat as a hemisphere broken into approximately
0012      C  equal area square elements ( similar to an Igloo ). A given relative
0013      C  threat is determined by calculating the Theta and Phi angles measured
0014      C  from the origin to the given element. The probability of this
0015      C  relative threat occuring is equal to the ratio of the element's
0016      C  surface area to the total surface area of the hemisphere.
0017      C
0018      C  The probability is further modified to account for the orbital motion
0019      C  of the space station. The velocity of the meteoroid relative to the
0020      C  space station is calculated as well as the flux focusing factor. Ted
0021      C  Hansen developed the flux focusing factor.
0022      C
0023      C  As a first order approximation the meteoroids are assumed to all have
0024      C  a absolute velocity equal to the average meteoroid velocity. This
0025      C  ignores the actual velocity distribution but appears to give reasonable
0026      C  results while greatly simplifying the analysis.
0027      C
0028      C  All the calculations are done in double precision but stored as single
0029      C  precision in the Threat array.
0030      C
0031      C  Note:  for variables included in the common block see main listing
0032      C
0033      C  Variable List
0034      C
0035      C      alpha = angle between the space station velocity vector and the
0036      C              relative threat vector, radians
0037      C      alt = altitude of the space station less 100km atmosphere, km
0038      C      answer = character string representing user input to a given question
0039      C      ar = ratio of a given element's surface area to the hemisphere surface
0040      C              area
0041      C      chi = pi - gamma, radians
0042      C      del = the delta phi angle, equal for each ring in the hemisphere,
0043      C              radians
0044      C      dphi = the angle measured from the z axis to the relative threat vector
0045      C              double precision, radians
0046      C      dpi = double precision pi
0047      C      dtheta = the angle measured from the space station velocity vector
0048      C              to the relative threat vector, double precision , radians
0049      C      gamma = half angle of the shaded cone , radians
0050      C      h = vertical height of a given ring
0051      C      l1 = intermediate variable, radians
0052      C      l2 =      "      "      "
0053      C      ic = counter
0054      C      prob = probability of a given relative threat occuring
0055      C      re = earth's radius (including 100 km atmosphere), km
0056      C      sf = portion of space station exposed to meteoroids
0057      C      sum = running sum of the probabilities
0058      C      tinc = theta increment in a given ring, radians
0059      C      tmf = flux focusing factor
0060      C      vm = velocity of the meteoroid, km/sec
0061      C      vr = velocity of the meteoroid relative to the space station , km/sec
0062      C      vs = orbital velocity of the space station, km/sec
0063      C      vstar = intermediate variable, km/sec

```

```

0064      C                               D180-30550-4
0065      C
0066      C   Array list
0067      C
0068      C   area = area of the elements in a ring
0069      C   dthreat = threat array in double precision
0070      C   nel = number of elements in a ring
0071      C   threat = array containing the relative theta & phi angles,
0072      C               the realtive velocity and the relative threat probability
0073      C
0074      C
0075      C       INCLUDE 'COMMON1.BLK'
0094      C
0095      C       DIMENSION NEL(50)
0096      C
0097      C       REAL*8 DPI
0098      C
0099      C       PARAMETER ( DPI=3.141592653589793238D0 )
0100      C
0101      C       CHARACTER*90 ANSWER
0102      C
0103      C       REAL*8 ALPHA,ALT,AR,CHI,DEL,DPHI,DTHETA,GAMMA,H,L1,L2,PROB,RE
0104      C       1      SF,SUM,TMF,VM,VR,VS,VSTAR,AREA(50),DTHREAT(4,1000)
0105      C
0106      C   Set the station and meteoroid velocity
0107      C
0108      C       VS=7.50D0
0109      C       VM=20.0D0
0110      C
0111      C   Set the Earth radius and Space Station altitude
0112      C
0113      C       RE=6478.0D0
0114      C       ALT=500.0D0-100.0D0
0115      C
0116      C   Read in the threat case to be run
0117      C
0118      C       40 WRITE ( 6,50 )
0119      C       50 FORMAT ( 1X,'NUMBER OF UNIFORM METEOROID THREATS ?',
0120      C           1      /5X,'1 - 84',/5X,'2 - 146<CR>',/5X,'3 - 232',
0121      C           2      /5X,'4 - 329',/1X,'ANSWER (1-4) >','$)
0122      C
0123      C       READ ( 5,60 ) ANSWER
0124      C       60 FORMAT (A)
0125      C
0126      C       IF ( ANSWER(1:2).EQ.' ' ) THEN
0127      C           IANS=2
0128      C       ELSE
0129      C           READ ( ANSWER(1:80),70 ) IANS
0130      C       70   FORMAT ( BN,I6 )
0131      C       END IF
0132      C
0133      C   Check if the threat case is within the given range
0134      C
0135      C       IF ( IANS.GT.4 .OR. IANS.LT.1 ) THEN
0136      C           WRITE ( 6,80 )
0137      C       80   FORMAT ( 'O','ANSWER OUTSIDE OF RANGE ' )
0138      C           GO TO 40
0139      C       END IF
0140      C
0141      C   Determine the half angle of the shaded cone,
0142      C   equation is from JSC-30000
0143      C
0144      C       GAMMA=DASIN(RE/(RE+ALT))

```

```

0145      CHI=DPI-GAMMA
0146      C
0147      C Calculate the number of uniform rings in the hemisphere, and the
0148      C delta phi angle for each ring
0149      C
0150      NR=6+(IANS-1)*2
0151      DEL=CHI/NR
0152      C
0153      C Determine the number elements and their area in the first ring
0154      C
0155      NEL(1)=IDNINT(2.0D0*DPI/DEL)
0156      H=DSIN(DEL)
0157      AREA(1)=2.0D0*DPI*H/NEL(1)
0158      C
0159      C Do the same for all the other rings holding the area approximately
0160      C equal.
0161      C
0162      DO 100 I=2,NR
0163      C
0164      H=ABS(DCOS(CHI-I*DEL)-DCOS(CHI-(I-1)*DEL))
0165      NEL(I)=IDNINT(2.0D0*DPI*H/AREA(1))
0166      AREA(I)=2.0D0*DPI*H/NEL(I)
0167      C
0168      100 CONTINUE
0169      C
0170      C Determine the number of elements ( equal to the number of threats )
0171      C
0172      NT=0
0173      DO 200 I=1,NR
0174      NT=NT+NEL(I)
0175      200 CONTINUE
0176      C
0177      C Each element represents a given relative threat direction. For each
0178      C element determine the location of it's C.G. . Also calculate the
0179      C relative meteoroid velocity for this threat and the flux focusing
0180      C factor. Store the results in the Threat array.
0181      C
0182      IC=0
0183      C
0184      DO 400 I=1,NR
0185      C
0186      C Calculate the theta increment in the ring
0187      C
0188      TINC=2.0D0*DPI/NEL(I)
0189      C
0190      C Calculate phi for the C.G., constant for each element in the ring
0191      C
0192      L1=DPI/2.0+(I-1)*DEL-CHI
0193      L2=L1+DEL
0194      DPFI=DACOS(0.50D0*(DSIN(L1)+DSIN(L2)))
0195      C
0196      C Calculate the probability of the threat occuring, constant for each
0197      C element in the ring
0198      C
0199      AR=AREA(I)/4.0D0/DPI
0200      C
0201      C Evaluate each element
0202      C
0203      DO 300 J=1,NEL(I)
0204      C
0205      IC=IC+1
0206      C
0207      C Determine the theta of the C.G.

```

D180-30550-4

```

0208 C
0209 DTHETA=TINC/2.0D0+(J-1)*TINC
0210 C
0211 C Calculate the relative meteoroid velocity and the flux focusing factor
0212 C
0213 ALPHA=DACOS(DSIN(DPHI)*DCOS(DTHETA))
0214 VSTAR=DSQRT(VM**2-VS**2*(DSIN(ALPHA)**2))
0215 VR=VSTAR+VS*DCOS(ALPHA)
0216 C
0217 TMF=VR**3/VSTAR/VM**2
0218 C
0219 C Apply the focusing factor to the probability
0220 C
0221 PROB=AR*TMF
0222 C
0223 C Store the values in the Threat array
0224 C
0225 DTHREAT(1,IC)=DTHETA
0226 DTHREAT(2,IC)=DPHI
0227 DTHREAT(3,IC)=VR
0228 DTHREAT(4,IC)=PROB
0229 C
0230 300 CONTINUE
0231 C
0232 400 CONTINUE
0233 C
0234 C Store Dthreat array in Threat
0235 C
0236 DO 600 I=1,NT
0237 THREAT(1,I)=DTHREAT(1,I)
0238 THREAT(2,I)=DTHREAT(2,I)
0239 THREAT(3,I)=DTHREAT(3,I)
0240 THREAT(4,I)=DTHREAT(4,I)
0241 600 CONTINUE
0242 C
0243 C Finished
0244 C
0245 RETURN
0246 C
0247 END

```

```

0001      C                                D180-30550-4
0002      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003      C
0004      SUBROUTINE DATA
0005      C
0006      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007      C
0008      C
0009      C Data reads in the supertab universal file and places the
0010      C element and nodal data in the global element array.
0011      C
0012      C
0013      C note: for variables in common block see main listing
0014      C
0015      C variable list
0016      C
0017      C dline = character string representing all the data contained
0018      C          on one line of the supertab universal file
0019      C eid = element identification
0020      C ig = grid id
0021      C igk = id of grid point in location k
0022      C ival = value contained in the first 6 spaces of dline
0023      C ig1 = id for element grid point 1
0024      C ig2 = " " " " " " 2
0025      C ig3 = " " " " " " 3
0026      C k = grid location pointer
0027      C kh = largest value of k
0028      C kl = smallest value of k
0029      C kp = previous value of k
0030      C ngrids = number of grids
0031      C pid = element property id
0032      C x = global x position , Meters
0033      C y = " y " , "
0034      C z = " z " , "
0035      C
0036      C array list
0037      C
0038      C grid = working array containing grid point locations in global
0039      C coordinate system
0040      C idg = working array containing grid point id's
0041      C
0042      C
0043      C
0044      CHARACTER*90 DLINE
0045      C
0046      INCLUDE 'COMMON1.BLK'
0047      C
0048      DIMENSION GRID(3,ISIZE*3)
0049      C
0050      INTEGER*4 IG1,IG2,IG3,IG,IGK,IVAL,IDG(ISIZE),K,KH,KL,EID,PID
0051      C
0052      C Intialize counters
0053      C
0054      C NELM = 0
0055      C NGRIDS = 0
0056      C
0057      C Read line of data from Supertab file , if end of file is reached
0058      C continue with processing
0059      C
0060      10 READ ( 2,20,END=100 )DLINE
0061      20 FORMAT ( A )
0062      C
0063      C Read first 6 characters of dline as an integer , if error then read

```

D180-30550-4

```

0082 C next line
0083 C
0084 READ (DLINE(1:6),30,ERR=10) IVAL
0085 30 FORMAT(BN,I6)
0086 C
0087 C Check if ival = -1 , which indicates the start of a dataset, if not
0088 C read another line until -1 is found
0089 C
0090 IF ( IVAL.NE.-1 ) GO TO 10
0091 C
0092 C -1 found , read next line of data, then read first six characters
0093 C if error on read, then read until end of dataset is found
0094 C
0095 READ ( 2,20 ) DLINE
0096 READ ( DLINE(1:6),30,ERR=90 ) IVAL
0097 C
0098 C First 6 characters are the dataset id, check if this is the nodal data
0099 C
0100 IF ( IVAL.EQ.15 ) THEN
0101 C
0102 C This is the nodal data , read data and look for the -1
0103 C that indicates the end the dataset has been reached
0104 C
0105 40 READ ( 2,20 ) DLINE
0106 READ ( DLINE(1:6),30 ) IVAL
0107 C
0108 C Check if bottom of dataset , if so start looking for next dataset
0109 C
0110 IF ( IVAL.EQ.-1 ) GO TO 10
0111 C
0112 C If not read grid id and location
0113 C
0114 READ ( DLINE(1:90),50 ) IG,X,Y,Z
0115 50 FORMAT ( BN,I10,30X,3E13.5 )
0116 C
0117 C Increment grid counter
0118 C
0119 NGRIDS=NGRIDS+1
0120 C
0121 C Place data in appropriate array
0122 C
0123 IDG(NGRIDS)=IG
0124 GRID(1,NGRIDS)=X
0125 GRID(2,NGRIDS)=Y
0126 GRID(3,NGRIDS)=Z
0127 C
0128 C Read next data line
0129 C
0130 GO TO 40
0131 C
0132 ELSE
0133 C
0134 C Is this the element dataset ?
0135 C
0136 IF ( IVAL.EQ.71 ) THEN
0137 C
0138 C It is the element, read next line of data, coonstantly checking for
0139 C end of dataset
0140 C
0141 60 READ ( 2,20 ) DLINE
0142 READ ( DLINE(1:6),30 ) IVAL
0143 IF ( IVAL.EQ.-1 ) GO TO 10
0144 C

```

D180-30550-4

```

0145 C Read eid and pid
0146 C
0147 READ ( DLINE(1:90),70 )EID,PID
0148 70 FORMAT ( BN,I10,20X,I10 )
0149 C
0150 C Read next line, and read ig1,ig2,ig3 off it
0151 C
0152 READ ( 2,20 ) DLINE
0153 READ ( DLINE(1:90),80 )IG1,IG2,IG3
0154 80 FORMAT( BN,3I10 )
0155 C
0156 C Increment element counter
0157 C
0158 NELM=NELM+1
0159 C
0160 C Place data in appropriate array
0161 C
0162 ID(1,NELM)=EID
0163 ID(2,NELM)=PID
0164 ID(3,NELM)=IG1
0165 ID(4,NELM)=IG2
0166 ID(5,NELM)=IG3
0167 C
0168 C Read in next data line
0169 C
0170 GO TO 60
0171 C
0172 ELSE
0173 C
0174 C It isn't the nodal or element dataset , read lines until bottom of
0175 C dataset is found, or end of file is found , if error on read read next
0176 C line
0177 C
0178 90 READ ( 2,20 )DLINE
0179 READ ( DLINE(1:6),30,ERR=90,END=100 )IVAL
0180 IF ( IVAL.NE.-1 ) GO TO 90
0181 GO TO 10
0182 C
0183 END IF
0184 C
0185 END IF
0186 C
0187 C Check if element data was read in, if not write error message and stop
0188 C
0189 100 IF ( NELM.EQ.0 ) THEN
0190 WRITE ( 6,110 )
0191 110 FORMAT ( /1X,'NO ELEMENT DATA WAS READ IN ' )
0192 STOP
0193 END IF
0194 C
0195 C Check if nodal data was read in , if not write error message and stop
0196 C
0197 IF ( NGRIDS.EQ.0 ) THEN
0198 WRITE ( 6,120 )
0199 120 FORMAT ( /1X,'NO NODAL DATA WAS READ IN ' )
0200 STOP
0201 END IF
0202 C
0203 C Place information in global element array
0204 C
0205 C Loop thru all the elements
0206 C
0207 DO 400 I=1,NELM

```

```

0208 C
0209 C   Loop thru 3 grid points
0210 C
0211 C       DO 300 J=1,3
0212 C
0213 C   Initialize grid point  location pointers
0214 C
0215 C       KH=NGRIDS+1
0216 C       KL=1
0217 C       KP=0
0218 C
0219 C   Get grid id
0220 C
0221 C       IG=ID((3+(J-1)),I)
0222 C
0223 C   Make guess of grid location using binary search technique
0224 C
0225 C   210       K=(KL+KH)/2
0226 C
0227 C   Check if same location is being guessed again, if it is write
0228 C   error message and stop
0229 C
0230 C       IF ( KP.EQ.K ) THEN
0231 C           WRITE ( 6,220 ) IG
0232 C   220       FORMAT(/1X,'DATA NOT FOUND FOR NODE ',I5 )
0233 C           STOP
0234 C       END IF
0235 C
0236 C   Get grid id for location k
0237 C
0238 C       IGK=IDG(K)
0239 C
0240 C   If this is the correct location put data in array element
0241 C
0242 C       IF ( IGK.EQ.IG ) THEN
0243 C
0244 C   Loop thru 3 coordinates
0245 C
0246 C       DO 250 L=1,3
0247 C           ELEMENT((L+(J-1)*3),I)=GRID(L,K)
0248 C   250       CONTINUE
0249 C
0250 C   If not , reset location limits , guess again
0251 C
0252 C       ELSE
0253 C
0254 C   Guess too low in list , reset lower bound
0255 C
0256 C       IF ( IGK.LT.IG ) KL=K
0257 C
0258 C   Guess too high in list , reset upper bound
0259 C
0260 C       IF ( IGK.GT.IG ) KH=K
0261 C
0262 C   Reset holder
0263 C
0264 C       KP=K
0265 C
0266 C   Guess again
0267 C
0268 C       GO TO 210
0269 C
0270 C       END IF

```

```
0271 C
0272 C Next grid point
0273 C
0274 300 CONTINUE
0275 C
0276 C Next element
0277 C
0278 400 CONTINUE
0279 C
0280 C
0281 C finished , return
0282 C
0283 C
0284 RETURN
0285 C
0286 END
```

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE NORMAL
0005 C
0006 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007 C
0008 C
0009 C Normal calculates the unit normal vector for each element and stores
0010 C it in the element array.
0011 C
0012 C variable list
0013 C
0014 C an = i component of the normal vector
0015 C a12 = " " " " 12 "
0016 C a13 = " " " " 13 "
0017 C bn = j component of the normal vector
0018 C b12 = " " " " 12 "
0019 C b13 = " " " " 13 "
0020 C cn = k component of the normal vector
0021 C c12 = " " " " 12 "
0022 C c13 = " " " " 13 "
0023 C r = length of the normal vector
0024 C
0025 C
0026 C
0027 C INCLUDE 'COMMON1.BLK'
0046 C
0047 C Calculate unit normal vector for each element
0048 C
0049 C DO 100 I=1,NELM
0050 C
0051 C Calculate vector from grid point 1 to grid point 2
0052 C
0053 C A12=ELEMENT(4,I)-ELEMENT(1,I)
0054 C B12=ELEMENT(5,I)-ELEMENT(2,I)
0055 C C12=ELEMENT(6,I)-ELEMENT(3,I)
0056 C
0057 C Calculate vector from grid point 1 to grid point 3
0058 C
0059 C A13=ELEMENT(7,I)-ELEMENT(1,I)
0060 C B13=ELEMENT(8,I)-ELEMENT(2,I)
0061 C C13=ELEMENT(9,I)-ELEMENT(3,I)
0062 C
0063 C Calculate the normal vector, it is equal to the cross product of vector
0064 C 12 and vector 13
0065 C
0066 C AN=(B12*C13)-(C12*B13)
0067 C BN=(C12*A13)-(A12*C13)
0068 C CN=(A12*B13)-(B12*A13)
0069 C
0070 C Calculate the length of the normal vector
0071 C
0072 C R=SQRT(AN**2+BN**2+CN**2)
0073 C
0074 C The unit normal vector is equal to the normal vector divided by it's
0075 C length, place data in element array
0076 C
0077 C ELEMENT(10,I)=AN/R
0078 C ELEMENT(11,I)=BN/R
0079 C ELEMENT(12,I)=CN/R
0080 C
0081 C Next element

```

D180-30550-4

0082	C	
0083		100 CONTINUE
0084	C	
0085	C	Return
0086	C	
0087		RETURN
0088	C	
0089		END

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE CENTROID
0005 C
0006 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007 C
0008 C Centroid determines the location of the centroid for each element,
0009 C and stores it in the element array.
0010 C
0011 C Variable list
0012 C
0013 C xc = x location of the centroid
0014 C yc = y location of the centroid
0015 C zc = z location of the centroid
0016 C
0017 C array list
0018 C
0019 C x = array containing x values of the grid points
0020 C y = " " " y " " " " "
0021 C z = " " " z " " " " "
0022 C
0023 C
0024 DIMENSION X(3),Y(3),Z(3)
0025 C
0026 INCLUDE 'COMMON1.BLK'
0045 C
0046 C Determine c.g. for each element
0047 C
0048 DO 100 I=1,NELM
0049 C
0050 C Get grid point data from element array
0051 C
0052 DO 50 J=1,3
0053 C
0054 X(J)=ELEMENT((1+(J-1)*3),I)
0055 Y(J)=ELEMENT((2+(J-1)*3),I)
0056 Z(J)=ELEMENT((3+(J-1)*3),I)
0057 C
0058 C Next grid point
0059 C
0060 50 CONTINUE
0061 C
0062 C C.G. location is average of 3 grid point locations
0063 C
0064 XC=(X(1)+X(2)+X(3))/3.0
0065 YC=(Y(1)+Y(2)+Y(3))/3.0
0066 ZC=(Z(1)+Z(2)+Z(3))/3.0
0067 C
0068 C Put data in element array
0069 C
0070 ELEMENT(13,I)=XC
0071 ELEMENT(14,I)=YC
0072 ELEMENT(15,I)=ZC
0073 C
0074 C Next element
0075 C
0076 100 CONTINUE
0077 C
0078 C Return
0079 C
0080 RETURN
0081 C

```

0082

END

```

0001      C                               D180-30550-4
0002      C
0003      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0004      C
0005      C          SUBROUTINE AREA
0006      C
0007      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008      C
0009      C  Area calculates the surface area of each element.
0010      C
0011      C
0012      C  Variable list
0013      C
0014      C    atri = surface area of the element , sq-meters
0015      C    d1 = intermediate variable
0016      C    d2 =      "      "
0017      C    d3 =      "      "
0018      C
0019      C  Array list
0020      C
0021      C    x = working array containing the grid point x coordinates
0022      C    y =      "      "      "      "      "      "      y      "
0023      C    z =      "      "      "      "      "      "      z      "
0024      C
0025      C
0026      C          INCLUDE 'COMMON1.BLK'
0045      C
0046      C          DIMENSION X(3),Y(3),Z(3)
0047      C
0048      C  For each element calculate the surface area.
0049      C
0050      C          DO 100 I=1,NELM
0051      C
0052      C  Get the location of the 3 grid points in global coordinates.
0053      C
0054      C          DO 50 J=1,3
0055      C
0056      C              X(J)=ELEMENT((1+(J-1)*3),I)
0057      C              Y(J)=ELEMENT((2+(J-1)*3),I)
0058      C              Z(J)=ELEMENT((3+(J-1)*3),I)
0059      C
0060      C          50  CONTINUE
0061      C
0062      C  Calculate the surface area using the equation from the CRC Math Handbook
0063      C
0064      C          D1=Y(1)*(Z(2)-Z(3))-Z(1)*(Y(2)-Y(3))+(Y(2)*Z(3)-Z(2)*Y(3))
0065      C
0066      C          D2=Z(1)*(X(2)-X(3))-X(1)*(Z(2)-Z(3))+(Z(2)*X(3)-X(2)*Z(3))
0067      C
0068      C          D3=X(1)*(Y(2)-Y(3))-Y(1)*(X(2)-X(3))+(X(2)*Y(3)-Y(2)*X(3))
0069      C
0070      C          ATRI = 0.50 * SQRT( D1**2 + D2**2 + D3**2)
0071      C
0072      C  Save the area in the element array
0073      C
0074      C          ELEMENT(17,I)=ATRI
0075      C
0076      C  Next element
0077      C
0078      C  100 CONTINUE
0079      C
0080      C
0081      C          RETURN

```

0082
0083

C

END

D180-30550-4

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE RADIUS
0005 C
0006 CCCCC6CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007 C
0008 C
0009 C Radius calculates the maximum radius that will contain the three grid
0010 C points of each triangular element, with the center located at the
0011 C centroid.
0012 C
0013 C Variable list
0014 C
0015 C a(k) = component 1 of vector from c.g. to grid k of element i
0016 C b(k) = component 2 of vector from c.g. to grid k of element i
0017 C c(k) = component 3 of vector from c.g. to grid k of element i
0018 C i = counter from 1 thru number of elements (nelm)
0019 C j = counter from 1 thru 3 nodes per element
0020 C k = counter from 1 thru 3 components per vector
0021 C l = counter from 1 thru 3 vectors per element
0022 C r(i) = length of vectors from c.g. to grid
0023 C rmax = maximum radius (squared) that contains all three nodes
0024 C x(j) = x coordinate of Ith element grid j
0025 C y(j) = y coordinate of Ith element grid j
0026 C z(j) = z coordinate of Ith element grid j
0027 C xc = x coordinate of Ith element c.g.
0028 C yc = y coordinate of Ith element c.g.
0029 C zc = z coordinate of Ith element c.g.
0030 C
0031 C
0032 INCLUDE 'COMMON1.BLK'
0033 C
0034 DIMENSION A(3),B(3),C(3),R(3),X(3),Y(3),Z(3)
0035 C
0036 DO 10 I=1,NELM
0037 C
0038 Read coordinates of c.g. of element I
0039 C
0040 XC=ELEMENT(13,I)
0041 YC=ELEMENT(14,I)
0042 ZC=ELEMENT(15,I)
0043 C
0044 DO 20 J=1,3
0045 C
0046 Read coordinates of three grids of element I
0047 C
0048 X(J)=ELEMENT((1+(J-1)*3),I)
0049 Y(J)=ELEMENT((2+(J-1)*3),I)
0050 Z(J)=ELEMENT((3+(J-1)*3),I)
0051 C
0052 20 CONTINUE
0053 C
0054 DO 30 K=1,3
0055 C
0056 Calculate vector from c.g. to each grid
0057 C
0058 A(K)=XC-X(K)
0059 B(K)=YC-Y(K)
0060 C(K)=ZC-Z(K)
0061 C
0062 30 CONTINUE
0063 C

```

D180-30550-4

```
0082          DO 40 L=1,3
0083      C
0084      C Calculate length (squared) of each vector from c.g. to each grid
0085      C
0086          R(L)=A(L)**2+B(L)**2+C(L)**2
0087      C
0088      40      CONTINUE
0089      C
0090      C Determine length of longest vector which will then be radius
0091      C
0092          RMAX=AMAX1( R(1),R(2),R(3) )
0093      C
0094          ELEMENT( 16,I )=RMAX
0095      C
0096      10 CONTINUE
0097      C
0098      C
0099          RETURN
0100      C
0101          END
```

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE BACKSIDE
0005 C
0006 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007 C
0008 C
0009 C Backside eliminates the elements not exposed to the relative threat
0010 C by virtue of the fact that their unit normal vector does not point
0011 C at the threat. It also generates the list of potentially exposed
0012 C elements.
0013 C
0014 C
0015 C Variable list
0016 C
0017 C a = i component of the unit threat vector
0018 C an = " " " " " normal "
0019 C b = j " " " " " threat "
0020 C bn = " " " " " normal "
0021 C c = k " " " " " threat "
0022 C cn = " " " " " normal "
0023 C dp = dot product of the unit normal vector and the unit threat
0024 C vector
0025 C ipx = element location in row 1 of the IEXP array
0026 C ipy = " " " " 2 " " " "
0027 C phi = relative threat angle measured from the Earth normal, radians
0028 C theta = relative threat angle measured from the station velocity
0029 C vector in the horizontal plane, radians
0030 C
0031 C
0032 C
0033 C INCLUDE 'COMMON1.BLK'
0052 C
0053 C For the first angle initialize the IEL, IEXP, & POINT arrays and
0054 C the NPE counter
0055 C
0056 C IF ( IT.NE.1 ) GO TO 20
0057 C
0058 C DO 10 I=1,NELM
0059 C
0060 C IEL(1,I)=.TRUE.
0061 C IEL(2,I)=.TRUE.
0062 C IEXP(1,I)=0
0063 C IEXP(2,I)=0
0064 C POINT(1,I)=0
0065 C POINT(2,I)=0
0066 C
0067 C 10 CONTINUE
0068 C
0069 C NPE=0
0070 C
0071 C Get the threat angles from the threat array
0072 C
0073 C 20 THETA=THREAT(1,IT)
0074 C PHI=THREAT(2,IT)
0075 C
0076 C Calculate the unit vector that points at the threat direction,
0077 C ( threat vector )
0078 C
0079 C A=SIN(PHI)*COS(THETA)
0080 C B=SIN(PHI)*SIN(THETA)
0081 C C=COS(PHI)

```

```

0082      C
0083      C   For each element , check if it is a backside element
0084      C
0085          DO 100 I=1,NELM
0086      C
0087      C   Get the unit normal vector from the element array
0088      C
0089          AN=ELEMENT(10,I)
0090          BN=ELEMENT(11,I)
0091          CN=ELEMENT(12,I)
0092      C
0093      C   Calculate the dot product of the unit normal vector and the unit
0094      C   threat vector
0095      C
0096          DP=A*AN+B*BN+C*CN
0097      C
0098      C   Save the dot product in the transform array
0099      C
0100          TRANSFORM(1,I)=DP
0101      C
0102      C   For the first angle case
0103      C
0104          IF ( IT.EQ.1 ) THEN
0105      C
0106      C   If the dot product is < 0 , then the element is a backside element
0107      C
0108          IF(DP.LE.0.01) THEN
0109      C
0110      C   It is backside , change IEL to false
0111      C
0112          IEL(1,I)=.FALSE.
0113          IEL(2,I)=.FALSE.
0114      C
0115          ELSE
0116      C
0117      C   It is potentially exposed , increase NPE by 1 and place I in IEXP
0118      C   location NPE
0119      C
0120          NPE=NPE+1
0121          IEXP(1,NPE)=I
0122          IEXP(2,NPE)=I
0123      C
0124          END IF
0125      C
0126      C   For all other cases , maintain the relative order of the IEXP list
0127      C   by adding exposed elements at the bottom and removing backside elements
0128      C   and shifting the list up . This will decrease the time required for
0129      C   sorting later in the code.
0130      C
0131          ELSE
0132      C
0133      C   If the dot product is < 0 then the element is a backside element
0134      C
0135          IF(DP.LE.0.01) THEN
0136      C
0137      C   It is a backside element , check if it was one in the previous case
0138      C
0139          IF ( IEL(1,I) ) THEN
0140      C
0141      C   It was not a backside element in the previous case, change IEL to
0142      C   false
0143      C
0144          IEL(1,I)=.FALSE.

```

```

0145             IEL(2,I)=.FALSE.
0146 C
0147 C Now, remove element I from the IEXP list , the list is not in order
0148 C
0149 C
0150 C Get element I location in IEXP from the pointer array
0151 C
0152             IPX=POINT(1,I)
0153             IPY=POINT(2,I)
0154 C
0155 C Remove element I and shift the list
0156 C
0157 C Shift the x list first, and reset the pointer list
0158 C
0159             DO 40 J=IPX,NPE
0160 C
0161                 IEXP(1,J)=IEXP(1,(J+1))
0162                 POINT(1,IEXP(1,J+1))=J
0163 C
0164 40             CONTINUE
0165 C
0166 C Shift the y list, and reset the pointer list
0167 C
0168             DO 45 J=IPY,NPE
0169 C
0170                 IEXP(2,J)=IEXP(2,(J+1))
0171                 POINT(2,IEXP(2,J+1))=J
0172 C
0173 45             CONTINUE
0174 C
0175 C Reduce the number of potentially exposed elements by 1
0176 C
0177             NPE=NPE-1
0178 C
0179 C Reset pointer for the removed element to 0
0180 C
0181             POINT(1,I)=0
0182             POINT(2,I)=0
0183 C
0184             END IF
0185 C
0186 C The element was a backside element in the previous case ,
0187 C no need to do anything
0188 C
0189             ELSE
0190 C
0191 C The element is potentially exposed, check if it was exposed in the
0192 C previous case
0193 C
0194             IF ( IEL(1,I) ) THEN
0195 C
0196 C It was exposed in the previous case , make IEL true
0197 C
0198                 IEL(1,I)=.TRUE.
0199                 IEL(2,I)=.TRUE.
0200 C
0201             ELSE
0202 C
0203 C It wasn't exposed previously, change IEL to true, increase NPE by
0204 C 1, add element I to IEXP at location NPE, and save location in the
0205 C pointer array
0206 C
0207             IEL(1,I)=.TRUE.

```

```

0208             IEL(2,I)=.TRUE.
0209      C
0210             NPE=NPE+1
0211      C
0212             IEXP(1,NPE)=I
0213             IEXP(2,NPE)=I
0214      C
0215             POINT(1,I)=NPE
0216             POINT(2,I)=NPE
0217      C
0218             END IF
0219      C
0220             END IF
0221      C
0222             END IF
0223      C
0224      C   Next element
0225      C
0226      100 CONTINUE
0227      C
0228      C   Check if number of exposed elements is 0 , if so write message to screen
0229      C
0230             IF ( NPE .EQ. 0 ) THEN
0231                 THETA=TR*180./3.14159
0232                 WRITE ( 6 , 200 )IT
0233      200     FORMAT ( /1X,'NO EXPOSED ELEMENTS FOR THREAT 'I4 )
0234             END IF
0235      C
0236      C   Check if number of exposed elements is > than the number of elements
0237      C   if so , write error message and stop
0238      C
0239             IF ( NPE .GT. NELM ) THEN
0240                 WRITE ( 6 , 300 )NPE,NELM
0241      300     FORMAT ( /1X,'NUMBER OF EXPOSED ELEMENTS = ',I5,/
0242                 .      /1X,'NUMBER OF ELEMENTS = ',I5 )
0243             STOP
0244             END IF
0245      C
0246      C   Return
0247      C
0248             RETURN
0249      C
0250             END

```

```

0001 C
0002 CCCCC6CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE TRANS
0005 C
0006 CCCCC6CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007 C
0008 C
0009 C Trans transforms and projects the c.g. and grid points onto a plane
0010 C orthogonal to the relative threat vector.
0011 C
0012 C
0013 C note: for variables in the common block refer to the main listing for
0014 C definition
0015 C
0016 C Variable list
0017 C
0018 C cp = cosine of phi
0019 C ct = cosine of theta
0020 C sp = sin of phi
0021 C st = sin of theta
0022 C phi = relative threat angle measured from the Earth normal, radians
0023 C theta = realative threat angle measured from the station velocity
0024 C vector in the horizontal plane, radians
0025 C xc = element c.g. x coordinate
0026 C yc = element c.g. y coordinate
0027 C zc = element c.g. z coordinate
0028 C xtc = c.g. x coordinate transformed to orthogonal plane
0029 C ytc = c.g. y coordinate transformed to orthogonal plane
0030 C ztc = c.g. z coordinate transformed to orthogonal plane
0031 C x(j) = x coordinate of jth grid of element i
0032 C y(j) = y coordinate of jth grid of element i
0033 C z(j) = z coordinate of jth grid of element i
0034 C
0035 C
0036 C
0037 C INCLUDE 'COMMON1.BLK'
0056 C
0057 C
0058 C DIMENSION X(3),Y(3),Z(3),XT(3),YT(3),ZT(3)
0059 C
0060 C Get theta and phi from the threat array
0061 C
0062 C THETA=THREAT(1,IT)
0063 C PHI=THREAT(2,IT)
0064 C
0065 C Calculate the cosine and sin of theta and phi
0066 C
0067 C CT=COS(THETA)
0068 C ST=SIN(THETA)
0069 C CP=COS(PHI)
0070 C SP=SIN(PHI)
0071 C
0072 C Evaluate the potentially exposed elements
0073 C
0074 C DO 10 I=1,NPE
0075 C
0076 C Transform coordinates of c.g. of element onto orthogonal plane
0077 C
0078 C XC=ELEMENT( 13,IEXP(1,I) )
0079 C YC=ELEMENT( 14,IEXP(1,I) )
0080 C ZC=ELEMENT( 15,IEXP(1,I) )
0081 C

```

```

0082      XCT = XC*SP*CT + YC*SP*ST + ZC*CP
0083      YCT = -XC*ST + YC*CT
0084      ZCT = -XC*CP*CT - YC*CP*ST + ZC*SP
0085  C
0086      TRANSFORM( 2, IEXP(1,I) ) = XCT
0087      TRANSFORM( 3, IEXP(1,I) ) = YCT
0088      TRANSFORM( 4, IEXP(1,I) ) = ZCT
0089  C
0090      DO 20 J=1,3
0091  C
0092  C      Transform grid coordinates onto the orthogonal plane
0093  C
0094      X(J) = ELEMENT( (1+(J-1)*3) , IEXP(1,I) )
0095      Y(J) = ELEMENT( (2+(J-1)*3) , IEXP(1,I) )
0096      Z(J) = ELEMENT( (3+(J-1)*3) , IEXP(1,I) )
0097  C
0098      XT(J)=  XCT
0099      YT(J)= -X(J)*ST + Y(J)*CT
0100      ZT(J)= -X(J)*CP*CT - Y(J)*CP*ST + Z(J)*SP
0101  C
0102      20      CONTINUE
0103  C
0104      DO 30 K=1,3
0105  C
0106  C      Store transformed grid coordinates into transform array
0107  C
0108      TRANSFORM( (5+(K-1)*3) , IEXP(1,I) )=XT(K)
0109      TRANSFORM( (6+(K-1)*3) , IEXP(1,I) )=YT(K)
0110      TRANSFORM( (7+(K-1)*3) , IEXP(1,I) )=ZT(K)
0111  C
0112  C
0113      30      CONTINUE
0114  C
0115  C
0116      10 CONTINUE
0117  C
0118      RETURN
0119  C
0120      END

```

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE QSORT ( IVAR )
0005 C
0006 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007 C
0008 C Sort Reals
0009 C
0010 C IMPLEMENTATION OF WIRTH'S DOCUMENTED QUICKSORT ALGORITHM.
0011 C
0012 C GIVEN AN ARRAY OF REAL VALUES AND AN ARRAY OF POINTERS
0013 C WHICH POINT TO THE VALUES, REARRANGE THE POINTERS SUCH THAT
0014 C THEY POINT TO VALUES OF INCREASING VALUE
0015 C BASED ON VALUES IN POINT.
0016 C
0017 C-----INPUT-----
0018 C
0019 C VALUES ARRAY OF ELEMENT CG X COORDINATES TO BE SORTED
0020 C IEXP ARRAY OF ELEMENT ID'S IN DECENDING ORDER
0021 C NPE THE LAST POINTER IS AT POSITION POINT(NPE)
0022 C
0023 C-----OUTPUT-----
0024 C
0025 C IEXP ARRAY OF ELEMENT ID'S SORTED ACCORDING TO
0026 C ORDER OF ARRAY POINT(NPE)
0027 C THEY POINT TO INCREASING VALUES OF
0028 C CG X COORDINATES OF ELEMENTS
0029 C
0030 C-----VARIABLES-----
0031 C
0032 C TEMP ARRAY TO TEMPORARILY STORE ARRAY IEXP WHEN SORTING
0033 C IEXP BASED ON VALUES IN ARRAY POINT
0034 C IBEGIN THE FIRST POINTER IS AT POSITION POINT(IBEGIN)
0035 C STACK ARRAY FOR SCRATCH USE ( MUST BE NPE-IBEGIN+1
0036 C WORDS LONG)
0037 C
0038 INCLUDE 'COMMON1.BLK'
0057 C
0058 INTEGER*2 IBEGIN,IPOINT,II,JJ
0059 INTEGER*2 POINTER(ISIZE),STACK(2,15),TEMP(ISIZE)
0060 INTEGER*2 I,J,L,R,S,W
0061 C
0062 IBEGIN = 1
0063 C
0064 C INITIALIZE ARRAY POINTER FROM 1 TO NPE
0065 C
0066 DO 5 I = 1, NPE
0067 5 POINTER(I)=I
0068 C
0069 CD WRITE(ALTOUT,10) IBEGIN,NPE,(POINTER(I),I=IBEGIN,NPE)
0070 CD10 FORMAT(' IAC7SI,UNSORTED POINTER,IBEGIN,NPE ',2I5/(' ',10I4))
0071 S=1
0072 STACK(1,1)=IBEGIN
0073 STACK(2,1)=NPE
0074 C
0075 C TAKE FROM TOP OF STACK
0076 C
0077 40 CONTINUE
0078 L=STACK(1,S)
0079 R=STACK(2,S)
0080 S=S-1
0081 C

```

```

0082 C      SPLIT  KEY(L) .....KEY(R)
0083 C
0084      80  CONTINUE
0085          I=L
0086          J=R
0087          IPOINT=POINTER((L+R)/2)
0088 C
0089      120  CONTINUE
0090          II=POINTER(I)
0091          IF (TRANSFORM(IVAR, IEXP((IVAR-1), II)) .GT.
0092              .   TRANSFORM(IVAR, IEXP((IVAR-1), IPOINT))) THEN
0093 C
0094          I=I+1
0095          GO TO 120
0096      END IF
0097 C
0098      160  CONTINUE
0099          JJ=POINTER(J)
0100          IF (TRANSFORM(IVAR, IEXP((IVAR-1), IPOINT)) .GT.
0101              .   TRANSFORM(IVAR, IEXP((IVAR-1), JJ))) THEN
0102 C
0103          J=J-1
0104          GO TO 160
0105      END IF
0106 C
0107          IF (I.LE.J) THEN
0108              W=POINTER(I)
0109              POINTER(I)=POINTER(J)
0110              POINTER(J)=W
0111              I=I+1
0112              J=J-1
0113              IF (I.LE.J) GO TO 120
0114          END IF
0115 C
0116          IF (J-L.LT.R-I) THEN
0117              IF (I.LT.R) THEN
0118 C
0119 C      STACK REQUEST FOR SORTING RIGHT PARTITION
0120 C
0121              S=S+1
0122              STACK(1,S)=I
0123              STACK(2,S)=R
0124 C
0125              END IF
0126 C
0127 C      CONTINUE SORTING LEFT PARTITION
0128 C
0129              R=J
0130 C
0131              ELSE
0132 C
0133              IF (L.LT.J) THEN
0134 C
0135 C      STACK REQUEST FOR SORTING LEFT PARTITION
0136 C
0137              S=S+1
0138              STACK(1,S)=L
0139              STACK(2,S)=J
0140              END IF
0141 C
0142 C      CONTINUE SORTING RIGHT PARTITION
0143 C
0144          L=I

```

D180-30550-4

```

0145      END IF
0146      C
0147      IF(L.LT.R) GO TO 80
0148      C
0149      IF(S.NE.0) GO TO 40
0150      C
0151      CD      WRITE(ALTOUT,500) IBEGIN,NPE,(POINTER(I),I=IBEGIN,NPE)
0152      CD500  FORMAT(' IAC7SI, SORTED POINTER, IBEGIN, NPE ',2I5/(' ',10I4))
0153      C
0154      C      STORE ARRAY IEXP TEMPORARILY IN ARRAY TEMP
0155      C
0156      DO 180 I = 1 , NPE
0157      180 TEMP(I) = IEXP((IVAR-1),I)
0158      C
0159      C      SORT ARRAY IEXP BASED ON VALUES IN ARRAY POINTER
0160      C
0161      DO 200 I = 1 , NPE
0162      C
0163      INDEX = POINTER(I)
0164      IEXP((IVAR-1),I) = TEMP(INDEX)
0165      C
0166      200 CONTINUE
0167      C
0168      RETURN
0169      C
0170      END

```

```

0001 C
0002 C
0003 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0004 C
0005 SUBROUTINE SHADOW
0006 C
0007 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008 C
0009 C Shadow determines which elements are exposed to the current threat.
0010 C It only considers the elements that were not eliminated by the
0011 C BACKSIDE subroutine.
0012 C
0013 C Shadow works with the sorted lists contained in the IEXP array. These
0014 C lists were sorted on the transformed x & y distances of the element
0015 C centroids from the threat. An element can be shadowed by another
0016 C element only if it lies below it in the x sorted list. As a crude
0017 C test, it's centroid must also lay within the circle enclosing the
0018 C shadowing element. The origin of this circle is located at the
0019 C centroid of the shadowing element. This test is accounted for by
0020 C evaluating the element's position in the y sorted list with respect to
0021 C the shadowing elements position. If the element passes this test, a
0022 C precise test is performed to determine if the element's centroid lies
0023 C within the shadowing element's boundaries. These boundaries have been
0024 C projected onto the plane normal to the threat.
0025 C
0026 C
0027 C note: for variables in the common block refer to main listing for
0028 C definition
0029 C
0030 C Variable list
0031 C
0032 C direction = logical variable @used to determine which direction to
0033 C move in the y sorted list
0034 C el = element number of the potentially shadowed element
0035 C icr = interger value of rt/srmax
0036 C icy = " " " yd/srmax
0037 C ipx = position in the x sorted list of the shadowing element
0038 C ipxel = " " " " " " " " " pot. shadowed element
0039 C ipy = " " " y " " " " " shadowing element
0040 C itest = element number of the shadowing element
0041 C pos = current position in the y sorted list
0042 C rt = square of the distance from the shadowing elements centroid to
0043 C the potentially shadowed elements centroid
0044 C srmax = square of the radius of the circle enclosing the shadowing
0045 C element with the origin at the centroid
0046 C yd = square of the y distance from the shadowing elements centroid
0047 C to the potentially shadowed elements centroid
0048 C yc = transformed y coordinate of the shadowing elements centroid
0049 C y1 = " " " " " " " " " grid point 1
0050 C y2 = " " " " " " " " " " " 2
0051 C y3 = " " " " " " " " " " " 3
0052 C y4 = " " " " " " " " " potentially shadowed elements
0053 C centroid
0054 C zc = transformed z coordinate of the shadowing elements centroid
0055 C z1 = " " " " " " " " " grid point 1
0056 C z2 = " " " " " " " " " " " 2
0057 C z3 = " " " " " " " " " " " 3
0058 C z4 = " " " " " " " " " potentially shadowed elements
0059 C centroid
0060 C
0061 C
0062 C
0063 C

```

D180-30550-4

```

0064 C
0065 INCLUDE 'COMMON1.BLK'
0084 C
0085 INTEGER*2 ICR, ICY, IPX, IPXEL, IPY, ITEST, EL, POS
0086 C
0087 LOGICAL DIRECTION, INTRI
0088 C
0089 DIRECTION=.TRUE.
0090 C
0091 C Starting with the first element in the x sorted list determine if
0092 C it shades any other elements.
0093 C
0094 DO 100 I=1,NPE
0095 C
0096 C Get the shadowing element
0097 C
0098 ITEST=IEXP(1,I)
0099 C
0100 C
0101 C Get the centroid location and the square of the maximum radius.
0102 C
0103 YC=TRANSFORM(3,ITEST)
0104 ZC=TRANSFORM(4,ITEST)
0105 C
0106 SRMAX=ELEMENT(16,ITEST)
0107 C
0108 C Determine the shadowing elements location in the x&y sorted list.
0109 C
0110 IPX=POINT(1,ITEST)
0111 IPY=POINT(2,ITEST)
0112 C
0113 C Begin evaluating the elements in the y sorted list , starting at
0114 C the next element after the shadowing element.
0115 C
0116 POS=IPY+1
0117 C
0118 C Determine the potentially shadowed element.
0119 C
0120 20 EL=IEXP(2,POS)
0121 C
0122 C Determine the potentially shadowed element's location in the x
0123 C sorted list.
0124 C
0125 IPXEL=POINT(1,EL)
0126 C
0127 C Check if the potentially shadowed element is below the shadowing
0128 C element in the x sorted list. If not get the next potentially
0129 C shadowed element.
0130 C
0131 IF ( IPXEL .GT. IPX ) THEN
0132 C
0133 C Check if the potentially shadowed element is exposed. If not get the
0134 C next potentially shadowed element.
0135 C
0136 IF ( IEL(2,EL) ) THEN
0137 C
0138 C Since the element is exposed , get it's centroid location.
0139 C
0140 Y4=TRANSFORM(3,EL)
0141 Z4=TRANSFORM(4,EL)
0142 C
0143 C Determine the y distance from the shadowing elements centroid to the
0144 C potentially shadowed elements centroid.

```

D180-30550-4

```

0145 C
0146 C          YD=(YC-Y4)**2
0147 C
0148 C Calculate the integer value of the distance to the radius ratio.
0149 C
0150 C          ICY=YD/SRMAX
0151 C
0152 C Using the integer of the ratio , determine if the element is outside
0153 C of radius of the shadowing element in the y direction. If so, there is
0154 C no need to go further through the y sorted list in the present
0155 C direction. Either switch directions and get next element or get next
0156 C shadowing element.
0157 C
0158 C          IF ( ICY.NE.0 ) GO TO 40
0159 C
0160 C Determine the distance from the elements centroid to the shadowing
0161 C elements centroid and square it.
0162 C
0163 C
0164 C          RT=YD+(ZC-Z4)**2
0165 C
0166 C Determine the integer value of the distance to maximum radius ratio.
0167 C
0168 C          ICR=RT/SRMAX
0169 C
0170 C Use the integer value to determine if the element is inside the radius.
0171 C If not get the next element.
0172 C
0173 C          IF ( ICR .EQ. 0 ) THEN
0174 C
0175 C The elements centroid is inside the maximum radius, perform the
0176 C precision test. First get the shadowing elements grid points location.
0177 C
0178 C          Y1=TRANSFORM(6,ITEST)
0179 C          Z1=TRANSFORM(7,ITEST)
0180 C          Y2=TRANSFORM(9,ITEST)
0181 C          Z2=TRANSFORM(10,ITEST)
0182 C          Y3=TRANSFORM(12,ITEST)
0183 C          Z3=TRANSFORM(13,ITEST)
0184 C
0185 C Set the exposure logical flag equal to the negative of the logical
0186 C function test for inside the triangle.
0187 C
0188 C          IEL(2,EL)=.NOT.INTRI(Y1,Z1,Y2,Z2,Y3,Z3,Y4,Z4)
0189 C
0190 C          END IF
0191 C
0192 C          END IF
0193 C
0194 C          END IF
0195 C
0196 C Increment the y list position based on which direction we are
0197 C currently moving. Check that we do not point to a location outside
0198 C the list.
0199 C
0200 C          IF ( DIRECTION ) THEN
0201 C
0202 C          POS=POS+1
0203 C          IF ( POS.GT.NPE ) GO TO 40
0204 C
0205 C          ELSE
0206 C
0207 C          POS=POS-1

```

D180-30550-4

```

0208             IF ( POS.LT. 1 ) GO TO 40
0209 C
0210             END IF
0211 C
0212 C   Get the next potentially shaded element.
0213 C
0214             GO TO 20
0215 C
0216 C   Change the direction and continue with the new starting position.
0217 C   Unless we have gone in both directions. Then get next shadowing
0218 C   element.
0219 C
0220 40             IF ( DIRECTION ) THEN
0221 C
0222             DIRECTION=.FALSE.
0223 C
0224             POS=IPY-1
0225 C
0226             IF ( POS .LT. 1 ) THEN
0227 C
0228             DIRECTION=.TRUE.
0229 C
0230             GO TO 100
0231 C
0232             END IF
0233 C
0234             GO TO 20
0235 C
0236             ELSE
0237 C
0238             DIRECTION=.TRUE.
0239 C
0240             END IF
0241 C
0242 C   Get the next shadowing element.
0243 C
0244 100 CONTINUE
0245 C
0246 C   Finished
0247 C
0248             RETURN
0249 C
0250             END

```

```

0001 C
0002 C
0003 C
0004 LOGICAL FUNCTION INTRI(AX,AY, BX,BY, CX,CY, PX,PY)
0005 C
0006 C Return 1 if point P is inside triangle or 0 if outside.
0007 C Triangle has vertices at A,B and C. P,A,B,C are in the XY plane.
0008 C EPS is the desired limiting precision in distance units.
0009 C
0010 INTRI=.FALSE.
0011 EPS= 1.E-6
0012 C
0013 C Move origin to A.
0014 C
0015 PPX= PX - AX
0016 PPY= PY - AY
0017 BPX= BX - AX
0018 BPY= BY - AY
0019 CPX= CX - AX
0020 CPY= CY - AY
0021 C
0022 C Exit if B or C is too near A.
0023 C
0024 D= SQRT(BPX*BPX + BPY*BPY)
0025 IF(D .LT. EPS .OR. SQRT(CPX*CPX + CPY*CPY) .LT. EPS) RETURN
0026 C
0027 C Form base unit vector BU along AB.
0028 C
0029 BUX= BPX/D
0030 BUY= BPY/D
0031 C
0032 C Define F + if ABC is ccw or - if cw, exit if collinear.
0033 C
0034 F= CPY*BUX - CPX*BUY
0035 IF(ABS(F).LT.EPS) RETURN
0036 M= SIGN(1.,F)
0037 C
0038 C Form altitude unit vector H perpendicular to and on C side of base.
0039 C
0040 HX= -BUY*M
0041 HY= BUX*M
0042 C
0043 C Heights V and Y of C and P, resp.
0044 C
0045 V= HX*CPX + HY*CPY
0046 Y= HX*PPX + HY*PPY
0047 C
0048 C Outside triangle if P is below base or above C.
0049 C
0050 IF(Y.LT.0. .OR. Y.GT.V) RETURN
0051 C
0052 C On-base projections U and X of C and P, resp.
0053 C
0054 U= BUX*CPX + BUY*CPY
0055 X= BUX*PPX + BUY*PPY
0056 C
0057 C P is outside the triangle if X is not between the intersections of
0058 C the sides with a line through P parallel to the base.
0059 C
0060 IF(X.LT.U*Y/V .OR. X.GT. D - (D-U)*Y/V) RETURN
0061 C
0062 C By elimination, the remaining condition is within the triangle.
0063 C

```

D180-30550-4

```
0064      INTRI=.TRUE.  
0065      C  
0066      END
```

D180-30550-4

```

0001 C
0002 C
0003 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0004 C
0005 SUBROUTINE OUTPUT
0006 C
0007 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008 C
0009 C Output writes the pertinent data out to the output file for each
0010 C element and each relative threat case. The output is in unformatted
0011 C form.
0012 C
0013 C
0014 C variable list
0015 C
0016 C nexp = number of exposed elements
0017 C
0018 C
0019 C INCLUDE 'COMMON1.BLK'
0038 C
0039 C INTEGER*2 I,NEXP
0040 C
0041 C For the first threat case write out the global threat and element data
0042 C
0043 C IF ( IT.EQ.1 ) THEN
0044 C
0045 C Write out the analysis type,number of threats, and the total number of
0046 C elements
0047 C
0048 C WRITE ( 4 ) ITYPE,NT,NELM
0049 C
0050 C Write out the threat array
0051 C
0052 C DO 10 I=1,NT
0053 C WRITE ( 4 ) (THREAT(J,I),J=1,4)
0054 C 10 CONTINUE
0055 C
0056 C Write out the element id and pid data
0057 C
0058 C DO 20 I=1,NELM
0059 C WRITE ( 4 ) (ID(J,I),J=1,2)
0060 C 20 CONTINUE
0061 C
0062 C Write out the element surface area
0063 C
0064 C DO 30 I=1,NELM
0065 C WRITE ( 4 ) ELEMENT(17,I)
0066 C 30 CONTINUE
0067 C
0068 C END IF
0069 C
0070 C Determine the number of exposed elements
0071 C
0072 C NEXP=0
0073 C DO 40 I=1,NELM
0074 C IF ( IEL(1,I).AND.IEL(2,I) ) THEN
0075 C NEXP=NEXP+1
0076 C END IF
0077 C 40 CONTINUE
0078 C
0079 C Write out the threat case number and the number of exposed elements
0080 C
0081 C WRITE ( 4 ) IT,NEXP

```

D180-30550-4

```

0082  C
0083  C  For each exposed element write out the cosine of the impact angle
0084  C
0085      DO 50 I=1,NELM
0086          IF ( IEL(1,I).AND.IEL(2,I) ) THEN
0087              WRITE ( 4 ) I,TRANSFORM(1,I)
0088          END IF
0089      50 CONTINUE
0090  C
0091  C  Finished
0092  C
0093      RETURN
0094  C
0095  C
0096      END

```

COMMON1.BLK

```

C
C Common block for Geometry Ver 2.5
C
C   ielm = number of elements
C
C   PARAMETER (IELM=9000)
C
C   INTEGER*2 IT, ITYPE, NELM, NPE, NT,
1      IEXP(2, IELM), POINT(2, IELM)
C
C   INTEGER*4 ID(5, IELM)
C
C   REAL*4 ELEMENT(17, IELM), TRANSFORM(13, IELM), THREAT(5, 500)
C
C   LOGICAL IEL(2, IELM)
C
C   COMMON IT, ITYPE, NELM, NPE, NT,
1      ID, IEL, IEXP, ELEMENT, POINT, THREAT, TRANSFORM

```

This page left intentionally blank.

D180-30550-4

APPENDIX C

Source Code for RESPONSE Module

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C C
0004 C RESPONSE VER 2.0 5/25/87 C
0005 C C
0006 C BOEING AEROSPACE CO. C
0007 C C
0008 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0009 C
0010 C
0011 C Response is a code for producing the hypervelocity impact response
0012 C file for the BUMPER code. BUMPER calculates the probability of no
0013 C penetration for spacecraft exposed to meteroids and man-made orbital
0014 C debris. The code is limited to the case of aluminum spheres impacting
0015 C aluminum two plate structures. 30 layers of multi-layer insulation may
0016 C be included between the plates. The code also considers all impact
0017 C angles greater than 60 deg ( measured from the normal ) to be equivalent
0018 C to 60 deg.
0019 C
0020 C The code was developed under the NASA contract 'Space Station Integrated
0021 C Wall Design Guide and Penetration Control', by M.A. Wright and
0022 C A.R. Coronado.
0023 C
0024 C
0025 C
0026 C Variable list
0027 C
0028 C aincr = impact angle increment,deg
0029 C amax = maximum impact angle,deg
0030 C amin = minimum impact angle,deg
0031 C ang = impact angle,deg
0032 C angr = impact angle,radians
0033 C answer = character string representing user input
0034 C ctype = configuration type
0035 C 1- single plate
0036 C 2 - double plate
0037 C dia = projectile diameter,in
0038 C diam = " " ,cm
0039 C ic = case counter
0040 C initial = logical variable used to determine if current call to
0041 C diameter is the initial one for the current angle
0042 C itype = analysis type,1=space debris
0043 C 2=meteoroids
0044 C mli = logical variable used to determine if 30 layers of mli is
0045 C included
0046 C nang = number of angles to be considered
0047 C nvel = number of velocities to be considered
0048 C pfunc = penetration function
0049 C 1-original
0050 C 2-pen4
0051 C 3-regression
0052 C shthk = shield thickness,in
0053 C stand = shield stand-off,in
0054 C vele = impact velocity,ft/sec
0055 C velm = " " ,km/sec
0056 C vincr = velocity increment,km/sec
0057 C vmax = maximum velocity,km/sec
0058 C vmin = minimum velocity,km/sec
0059 C
0060 C
0061 C Array list
0062 C
0063 C bhard = array containing the Brinell hardness values for the

```

D180-30550-4

```

0064      C      current configuration
0065      C      c = array containing the speed of sound values for the current
0066      C      configuration, ft/sec
0067      C      dens = array containing the density values for the current
0068      C      configuration, lbs/in**3
0069      C      fsu = array containing the shear allowable stress values for the
0070      C      current configuration, psi
0071      C      ftu = array containing the ultimate tensile stress values for the
0072      C      current configuration, psi
0073      C      fy = array containing the yield stress values for the current
0074      C      configuration
0075      C      rtable = array containing the critical diameters for each case, angle
0076      C      and velocity
0077      C      shpv = array containing the shock projectile velocities for the
0078      C      current configuration
0079      C      wilkc = array containing the values for Wilkinson's constant for
0080      C      the current configuration, km/sec
0081      C
0082      C
0083      C
0084      C
0085      CHARACTER*20 ANSWER
0086      C
0087      DIMENSION RTABLE(70,50,10)
0088      C
0089      DIMENSION BHARD(3), C(3), DENS(3), FSU(3), FTU(3), FY(3), SHPV(3),
0090      1      WILKC(3)
0091      C
0092      INTEGER*2 CTYPE, IC, ITYPE, NANG, NVEL, PFUNC
0093      C
0094      LOGICAL INITIAL, METRIC, MLI
0095      C
0096      C      Initialize variables
0097      C
0098      IC=0
0099      PFUNC=0
0100      C
0101      C      Set the angle and velocity limits and increments
0102      C
0103      C      Amin must always =0
0104      C
0105      AMIN=0.0
0106      C
0107      AMAX=90.0
0108      AINCR=5.0
0109      C
0110      C      Determine the number of velocity and angle iterations
0111      C
0112      NANG = (AMAX-AMIN)/AINCR + 1
0113      C
0114      C      Write header to screen
0115      C
0116      CALL HEADER (ITYPE)
0117      C
0118      C      Set velocity values based on analysis type
0119      C
0120      C      Vmin must always = 0
0121      C
0122      VMIN=0.0
0123      C
0124      IF ( ITYPE.EQ.1 ) THEN
0125      VMAX=17.0
0126      VINCR=0.25

```

D180-30550-4

```

0127      ELSE
0128          VMAX=70.0
0129          VINCR=1.0
0130      END IF
0131  C
0132          NVEL=(VMAX-VMIN)/VINCR
0133  C
0134  C Increment the case counter,and determine the wall configuration
0135  C
0136      10 IC=IC+1
0137  C
0138          CALL INPUT ( CTYPE,IC,ITYPE,MLI,PFUNC,SHTHK,STAND,VWTHK,BHARD,
0139      1              C,DENS,FSU,FTU,FY,SHPV,WILKC )
0140  C
0141  C For the current configuration,determine the critical diameter
0142  C for each impact angle and velocity
0143  C
0144          DO 200 I=1,NANG
0145  C
0146  C Set the angle,in deg and radians
0147  C
0148          ANG = 0.0 + (I-1)*AINCR
0149  C
0150  C For angles > 60 deg,set ang=60
0151  C
0152          IF ( ANG .GT. 60.0 ) ANG=60.0
0153  C
0154  C Convert ang to radians
0155  C
0156          ANGR = ANG / 180.0 * 3.141592
0157  C
0158  C Set initial equal to true
0159  C
0160          INITIAL=.TRUE.
0161  C
0162          DO 100 J=1,NVEL
0163  C
0164  C Set the velocity in ft/sec and km/sec
0165  C
0166          VELM = J*VINCR
0167  C
0168  C Convert vel to ft/sec
0169  C
0170          VELE = VELM * 1.0E+05 / 2.54 / 12.0
0171  C
0172  C Determine the critical diameter, as a function of wall configuration
0173  C
0174          IF ( CTYPE.EQ.2 ) THEN
0175  C
0176  C For the original and regression penetration functions use DOUBLE
0177  C subroutines
0178  C
0179          IF ( PFUNC.EQ.1 .OR. PFUNC.EQ.3 ) THEN
0180              CALL DOUBLE ( ANGR,DIA,INITIAL,ITYPE,MLI,PFUNC,SHTHK,
0181      1                  STAND,VELE,VELM,VWTHK,BHARD,C,DENS,
0182      2                  FSU,FTU,FY,SHPV,WILKC )
0183  C
0184  C For Pen4 use the pen4 subroutine
0185  C
0186          ELSE IF ( PFUNC.EQ.2 ) THEN
0187              CALL PEN4 ( ANGR,BHARD,C,DENS,DIA,VWTHK,FY,FSU,FTU,
0188      1                  INITIAL,SHATTER,SHPV,SHTHK,STAND,VELE )
0189          END IF

```

D180-30550-4

```

0190      C
0191          ELSE
0192              CALL SINGLE ( ANGR,DIA,VELE,VWTHK,DENS,FTU )
0193          END IF
0194      C
0195      C Convert the diameter to cm
0196      C
0197          DIAM = DIA * 2.54
0198      C
0199      C Store the diameter in RTABLE
0200      C
0201          RTABLE(J,I,IC)=DIAM
0202      C
0203      100      CONTINUE
0204      C
0205      200 CONTINUE
0206      C
0207      C Determine if another case is to be run
0208      C
0209          WRITE ( 6,300 )
0210      300 FORMAT(/1X,'DO YOU WISH TO RUN ANOTHER CASE ? <CR>=YES : ','$)
0211          READ ( 5,310 ) ANSWER
0212      310 FORMAT ( A )
0213      C
0214          IF ( ANSWER(1:1).NE.'N' ) GO TO 10
0215      C
0216      C Write response information to output file
0217      C
0218          WRITE ( 8 ) ITYPE,IC
0219      C
0220          WRITE ( 8 ) NANG,AINCR
0221      C
0222          WRITE ( 8 ) NVEL,VINCR
0223      C
0224          DO 500 I=1,IC
0225              DO 450 J=1,NANG
0226                  DO 400 K=1,NVEL
0227                      WRITE ( 8 ) RTABLE(K,J,I)
0228      400          CONTINUE
0229      450          CONTINUE
0230      500 CONTINUE
0231      C
0232      C Close the output files
0233      C
0234          CLOSE ( UNIT=7,STATUS='KEEP')
0235          CLOSE ( UNIT=8,STATUS='KEEP')
0236      C
0237      END

```

```

0001      C
0002      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003      C
0004          SUBROUTINE HEADER (ITYPE)
0005      C
0006      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007      C
0008      C
0009      C
0010      C Header writes the header to the screen and determines the analysis
0011      C type.
0012      C
0013      C
0014      C
0015          INTEGER*2 ITYPE
0016      C
0017          CHARACTER*80 ANSWER
0018      C
0019          WRITE ( 6,10 )
0020      10 FORMAT ( /1X,'*****',//,2X,'RESPONSE VER 2.0',
0021      1          //,1X,'Last Update 5/25/87',//,'*****')
0022      C
0023      C Determine the analysis type
0024      C
0025      20 WRITE ( 6,30 )
0026      30 FORMAT ( /1X,'ANALYSIS TYPE ?',//,2X,'1-DEBRIS <CR>',//,2X,
0027      1          '2-METEORIDS',//,1X,'ANSWER 1 OR 2 : ', $)
0028      C
0029          READ ( 5,40 ) ANSWER
0030      40 FORMAT (A)
0031      C
0032          IF ( ANSWER(1:1).EQ.' ' ) THEN
0033              ITYPE=1
0034          ELSE
0035              READ ( ANSWER(1:80),50 ) ITYPE
0036      50      FORMAT ( BN,I4 )
0037          END IF
0038      C
0039          IF ( ITYPE.EQ.1 .OR. ITYPE.EQ.2 ) THEN
0040              CONTINUE
0041          ELSE
0042              WRITE ( 6,60 )
0043      60      FORMAT ( /1X,'INCORRECT INPUT' )
0044              GO TO 20
0045          END IF
0046      C
0047          RETURN
0048      C
0049          END

```

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE INPUT (CTYPE,IC,ITYPE,MLI,PFUNC,SHTHK,STAND,VWTHK,
0005 1 BHARD,C,DENS,FSU,FTU,FY,SHPV,WILKC )
0006 C
0007 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008 C
0009 C
0010 C Input reads in the physical properties file, determines the wall
0011 C configuration, and writes the configuration information out to the
0012 C properties output file.
0013 C
0014 C
0015 C Variable list
0016 C
0017 C answer = character string representing user input
0018 C ctype = configuration type
0019 C 1- single plate
0020 C 2- double plate
0021 C ds = Epson printer control for double space
0022 C ic = current case number
0023 C itype = analysis type
0024 C lunits = length unit,cm for metric, in for english
0025 C metric = a logical variable used to determine if the input is in
0026 C metric units
0027 C mli = logical variable used to determine if multi-layer insulation
0028 C is used between the shield and vessel wall
0029 C ml = length of the material menu
0030 C nc = material counter < 10
0031 C ofile1 = properties output file
0032 C ofile2 = response output file
0033 C pfunc = penetration function
0034 C 1- original
0035 C 2- pen4
0036 C 3- regression
0037 C pid = property id
0038 C shthk = shield thickness,in or cm
0039 C stand = shield stand-off distance,in or cm
0040 C tp = Epson printer control for top of page
0041 C vwthk = vessel wall thickness,in or cm
0042 C
0043 C Array list
0044 C
0045 C bhard = array containing the shield ,vessel wall, and projectile
0046 C Brinell hardness
0047 C c = array containing the shield, vessel wall, and projectile speed
0048 C of sound in ft/sec
0049 C constant = array containing the values of Wilkinson's constant for all
0050 C materials in the physical properties file, km/sec
0051 C dens = array containing the shield, vessel wall, and projectile density
0052 C in lbs/in**3
0053 C density = array containing the value of the density for all the materials
0054 C in lb/in**3
0055 C hard = array containing the values of the Brinell hardness for all the
0056 C materials
0057 C fsu = array containing the shield, vessel wall, and projectile shear
0058 C stress, psi
0059 C ftu = array containing the shield, vessel wall, and projectile ultimate
0060 C tensile stress, psi
0061 C fty = array containing the shield, vessel wall, and projectile tensile
0062 C yield stress, psi
0063 C mat = array containing the material number for the shield, vessel wall,

```

```

0064 C      and the projectile respectively
0065 C      material = array containing all of the material names for all of the
0066 C      materials in the physical properties file
0067 C      shear = array containing the values of the shear stress for all of the
0068 C      materials, psi
0069 C      shock = array containing the values of the shock projectile velocity
0070 C      for all of the materials, values obtained from Physics
0071 C      shpv = array containing the shield, vessel wall, all projectile shock
0072 C      projectile velocities
0073 C      sound = array containing the values of the speed of sound for all of
0074 C      the materials,ft/sec
0075 C      wilkc = array containing the shield, vessel wall, and projectile
0076 C      Wilkkinson's constant,km/sec
0077 C      yield = array containing the values of the yield stress for all the
0078 C      materials,psi
0079 C      ult = array containing the values of the ultimate tensile stress for
0080 C      all of the materials,psi
0081 C
0082 C
0083 C
0084 C      CHARACTER*1 DS,TP
0085 C      CHARACTER*2 LUNITS
0086 C      CHARACTER*12 MATERIAL(10)
0087 C
0088 C      CHARACTER*80 ANSWER,OF1,OF2
0089 C
0090 C      DIMENSION BHARD(3),C(3),DENS(3),FSU(3),FTU(3),FY(3),SHPV(3),
0091 C      .          WILKC(3)
0092 C
0093 C      DIMENSION DENSITY(10),CONSTANT(10),HARD(10),SHEAR(10),SHOCK(10),
0094 C      .          SOUND(10),ULT(10),YIELD(10)
0095 C
0096 C      INTEGER*2 CTYPE,IC,ITYPE,ML,NC,PFUNC,PID,MAT(3)
0097 C
0098 C      LOGICAL METRIC,MLI
0099 C
0100 C      SAVE
0101 C
0102 C      For the first case only, initialize the variables & read in the
0103 C      material properties file.
0104 C
0105 C      IF (IC.GT.1) GO TO 70
0106 C
0107 C      NC=1
0108 C      METRIC=.TRUE.
0109 C      MLI=.FALSE.
0110 C
0111 C      OPEN (UNIT=2,FILE='MAT.PRP',STATUS='OLD')
0112 C
0113 C      10 READ ( 2,20,END=30 )MATERIAL(NC),DENSITY(NC),YIELD(NC),
0114 C      1          ULT(NC),SHEAR(NC),CONSTANT(NC),SOUND(NC),
0115 C      2          SHOCK(NC),HARD(NC)
0116 C      20 FORMAT (A,BN,8E12.5)
0117 C
0118 C      ML=NC
0119 C      NC=NC+1
0120 C
0121 C      GO TO 10
0122 C
0123 C      30 CLOSE ( UNIT=2,STATUS='KEEP' )
0124 C
0125 C      Read in the output file names.
0126 C

```

```

0127      WRITE ( 6,40 )
0128      40 FORMAT (/1X,'OUTPUT FILENAME FOR WALL PROPERTIES ',
0129      .      ' <CR>=RESPONSE.SUM : ',,$)
0130      READ ( 5,'(A)' )ANSWER
0131      IF ( ANSWER(1:1) .EQ. ' ' ) ANSWER='RESPONSE.SUM'
0132      OFILE1=ANSWER
0133      C
0134      WRITE ( 6,50 )
0135      50 FORMAT (/1X,'OUTPUT FILENAME FOR RESPONSE TABLES ',
0136      .      ' <CR>=STATION.RSP : ',,$)
0137      READ ( 5,'(A)' )ANSWER
0138      IF ( ANSWER(1:1) .EQ. ' ' ) ANSWER='STATION.RSP'
0139      OFILE2=ANSWER
0140      C
0141      C Open the output files.
0142      C
0143      OPEN ( UNIT=7, FILE=OFILE1, STATUS='UNKNOWN' )
0144      OPEN ( UNIT=8, FILE=OFILE2, STATUS='UNKNOWN',FORM='UNFORMATTED' )
0145      C
0146      REWIND 7
0147      REWIND 8
0148      C
0149      C Determine the type of units for input.
0150      C
0151      WRITE ( 6,60 )
0152      60 FORMAT (/1X,'INPUT IN METRIC OR ENGLISH UNITS <CR>=METRIC : ',,$)
0153      READ ( 5,'(A)' )ANSWER
0154      IF ( ANSWER(1:1) .EQ. 'E' ) METRIC=.FALSE.
0155      C
0156      C Set the units and if they are metric convert the variables to metric.
0157      C
0158      70 IF ( METRIC ) THEN
0159          LUNITS='CM'
0160          SHTHK=SHTHK*2.54
0161          VWTHK=VWTHK*2.54
0162          STAND=STAND*2.54
0163      ELSE
0164          LUNITS='IN'
0165      END IF
0166      C
0167      C Set the property id to the current case number and display
0168      C
0169      PID=IC
0170      C
0171      WRITE ( 6,80 ) IC
0172      80 FORMAT (/1X,'PROPERTY ID NUMBER = ',I2 )
0173      C
0174      C Determine configuration type
0175      C
0176      90 WRITE ( 6,100 )
0177      100 FORMAT ( /1X,'CONFIGURATION TYPE',/,5X,'1- SINGLE PLATE ',/,
0178      1      5X,'2- DOUBLE PLATE <CR>',/,1X,'ANSWER (1 or 2) : ',,$)
0179      READ ( 5,'(A)' ) ANSWER
0180      IF ( ANSWER(1:1) .EQ. ' ' ) THEN
0181          CTYPE=2
0182      ELSE
0183          READ ( ANSWER(1:12),110,ERR=90 ) CTYPE
0184      110      FORMAT ( BN,I4 )
0185      END IF
0186      C
0187      C Check that the input was correct
0188      C
0189      IF ( CTYPE.LT.1 .OR. CTYPE.GT.2 ) THEN

```

```

0190      WRITE ( 6,120 )
0191      120  FORMAT ( /1X,'INCORRECT INPUT' )
0192      GO TO 90
0193      END IF
0194      C
0195      C For single plate configuration skip down to the vessel wall material
0196      C
0197      IF ( CTYPE.EQ.1 ) GO TO 290
0198      C
0199      C Determine which double wall penetration function to use for the first
0200      C double wall case only
0201      C
0202      IF ( PFUNC.GT.0 ) GO TO 150
0203      C
0204      130 WRITE ( 6,140 )
0205      140 FORMAT ( /1X,'PENETRATION FUNCTION ',/,5X,'1-ORIGINAL <CR>',/,
0206      1      5X,'2-PEN4',/,5X,'3-REGRESSION',/,1X,'ANSWER (1-3) : ', $)
0207      READ ( 5,'(A)' ) ANSWER
0208      IF ( ANSWER(1:1).EQ.' ' ) THEN
0209          PFUNC=1
0210      ELSE
0211          READ ( ANSWER(1:80),'(BN,I4)',ERR=130 ) PFUNC
0212      END IF
0213      C
0214      C Check Input
0215      C
0216      IF ( PFUNC.LT.1 .OR. PFUNC.GT.3 ) GO TO 130
0217      C
0218      C Determine the shield material.
0219      C
0220      150 WRITE ( 6,160 )
0221      160 FORMAT (/1X,'SHIELD MATERIAL ')
0222      C
0223      C Write out the material list.
0224      C
0225      DO 180 I=1,ML
0226      C
0227          WRITE ( 6,170 ) I,MATERIAL(I)
0228      170  FORMAT ( 3X,I2,'- ',A )
0229      C
0230      180 CONTINUE
0231      C
0232      C For the initial case, set the material default number equal to one.
0233      C For all other cases use the previous shield material number as the
0234      C default. If an error is detected on the read, repeat the process.
0235      C
0236      IF ( IC .EQ. 1 ) THEN
0237      190  WRITE ( 6,220 )
0238          READ ( 5,'(A)' ) ANSWER
0239          IF ( ANSWER(1:1).EQ.' ' ) ANSWER='1'
0240          READ (ANSWER(1:4),200,ERR=190)MAT(1)
0241      200  FORMAT(BN,I4)
0242      ELSE
0243      210  WRITE ( 6,230 ) MAT(1)
0244          READ ( 5,'(A)' ) ANSWER
0245          IF ( ANSWER(1:1).NE.' ' ) THEN
0246              READ ( ANSWER(1:4),200,ERR=210 ) MAT(1)
0247          END IF
0248      ENDIF
0249      220 FORMAT (1X,'SELECT MATERIAL NUMBER <CR>=1 : ', $)
0250      230 FORMAT (1X,'SELECT MATERIAL NUMBER <CR>=',I2,' : ', $)
0251      C
0252      C Check that the value read in is contained in the list.

```

```

0253 C
0254 IF ( MAT(1) .LT.1 .OR. MAT(1).GT. ML ) GO TO 150
0255 C
0256 C Determine the shield thickness. For the initial case there is no default,
0257 C for all other cases use the previous value as the default.
0258 C
0259 IF ( IC.EQ.1 ) THEN
0260 240 WRITE ( 6,270 ) LUNITS
0261 READ ( 5,*,ERR=240 ) SHTHK
0262 ELSE
0263 250 WRITE ( 6,280 ) LUNITS,SHTHK
0264 READ ( 5,'(A)' ) ANSWER
0265 IF ( ANSWER(1:1).NE.' ' ) THEN
0266 READ ( ANSWER(1:12),260,ERR=250 ) SHTHK
0267 260 FORMAT(BN,E12.0)
0268 END IF
0269 END IF
0270 270 FORMAT ( /1X,'SHIELD THICKNESS (' ,A,' ) = : ', $)
0271 280 FORMAT ( /1X,'SHIELD THICKNESS (' ,A,' ) = ',F8.3,' : ', $)
0272 C
0273 C Determine the vessel wall material. Use the same technique as used
0274 C to determine the shield material.
0275 C
0276 290 WRITE ( 6,300 )
0277 300 FORMAT (/1X,'VESSEL WALL MATERIAL ' )
0278 C
0279 DO 310 I=1,ML
0280 WRITE ( 6,170 ) I,MATERIAL(I)
0281 310 CONTINUE
0282 C
0283 IF ( IC.EQ.1 ) THEN
0284 320 WRITE ( 6 ,220 )
0285 READ ( 5,'(A)' ) ANSWER
0286 IF ( ANSWER(1:1) .EQ. ' ' ) ANSWER='1'
0287 READ ( ANSWER (1:4),200,ERR=320 ) MAT(2)
0288 ELSE
0289 330 WRITE ( 6,230 ) MAT(2)
0290 READ ( 5,'(A)' ) ANSWER
0291 IF ( ANSWER(1:1).NE.' ' ) THEN
0292 READ ( ANSWER(1:4),200,ERR=330 ) MAT(2)
0293 END IF
0294 END IF
0295 C
0296 IF ( MAT(2).LT.1 .OR. MAT(2).GT.ML ) GO TO 290
0297 C
0298 C Determine the vessel wall thickness.
0299 C
0300 IF ( IC .EQ. 1 ) THEN
0301 340 WRITE ( 6,360 ) LUNITS
0302 READ ( 5,*,ERR=340 ) VWTHK
0303 ELSE
0304 350 WRITE ( 6,370 ) LUNITS,VWTHK
0305 READ ( 5,'(A)' ) ANSWER
0306 IF ( ANSWER(1:1).NE.' ' ) THEN
0307 READ (ANSWER(1:12),260,ERR=350) VWTHK
0308 END IF
0309 END IF
0310 360 FORMAT (/1X,'VESSEL WALL THICKNESS (' ,A,' ) = : ', $)
0311 370 FORMAT (/1X,'VESSEL WALL THICKNESS (' ,A,' ) = ',F8.3,' : ', $)
0312 C
0313 C For single plate configuration skip stand-off and mli
0314 C
0315 IF ( CTYPE.EQ.1 ) GO TO 430

```

```

0316 C
0317 C Determine the shield stand-off distance.
0318 C
0319 IF ( IC.EQ.1 ) THEN
0320 380 WRITE ( 6,400 ) LUNITS
0321 READ ( 5,*,ERR=380 ) STAND
0322 ELSE
0323 390 WRITE ( 6,410 ) LUNITS,STAND
0324 READ ( 5,'(A)' ) ANSWER
0325 IF ( ANSWER(1:1).NE.' ' ) THEN
0326 READ ( ANSWER(1:12),260,ERR=390 ) STAND
0327 END IF
0328 END IF
0329 400 FORMAT ( /1X,'SHIELD STAND-OFF (' ,A,' ) = : ', $ )
0330 410 FORMAT ( /1X,'SHIELD STAND-OFF (' ,A,' ) = ',F8.3,' : ', $ )
0331 C
0332 C Determine if MLI is to be included, but not for the pen4 penetration
0333 C function
0334 C
0335 IF ( PFUNC.EQ.2 ) GO TO 430
0336 C
0337 WRITE ( 6,420 )
0338 420 FORMAT (/1X,'INCLUDE 30 LAYERS OF MLI BETWEEN PLATES <CR>=YES : ',
0339 1 $ )
0340 READ ( 5,'(A)' ) ANSWER
0341 IF ( ANSWER(1:1).EQ.' ' .OR. ANSWER(1:1).EQ.'Y' ) MLI=.TRUE.
0342 C
0343 C Set the projectile material property based on analysis type
0344 C For debris use 100-0 al, for meteoroids only density is important use
0345 C 0.50 g/cc .
0346 C
0347 430 IF ( ITYPE.EQ.1 ) THEN
0348 BHARD(3)=23.0
0349 C(3)=16550.0
0350 DENS(3)=0.098
0351 FSU(3)=5000.0
0352 FTU(3)=13000.0
0353 FY(3)=9000.0
0354 SHPV(3)=1.345
0355 WILKC(3)=.126
0356 ELSE
0357 BHARD(3)=0.0
0358 C(3)=0.0
0359 DENS(3)=0.50/27.705
0360 FSU(3)=0.0
0361 FTU(3)=0.0
0362 FY(3)=0.0
0363 SHPV(3)=0.0
0364 WILKC(3)=0.0
0365 END IF
0366
0367 C
0368 C Write out the inputed values to the properties output file.
0369 C
0370 TP=CHAR(12)
0371 DS=CHAR(10)
0372 C
0373 WRITE ( 7,440 ) TP
0374 440 FORMAT ( A,'RESPONSE VER 2.0 ' )
0375 C
0376 IF ( ITYPE.EQ.1 ) THEN
0377 WRITE ( 7,450 ) DS
0378 ELSE

```

```

0379         WRITE ( 7,460 ) DS
0380     END IF
0381     450 FORMAT (A,'MAN-MADE ORBITAL DEBRIS ANALYSIS' )
0382     460 FORMAT (A,'METEOROID ANALYSIS ' )
0383 C
0384         WRITE ( 7,465 ) DS,PID
0385     465 FORMAT(A,'PROPERTY ID ',I4)
0386 C
0387         IF ( CTYPE.EQ.1 ) THEN
0388             WRITE ( 7,470 )DS
0389         ELSE
0390             WRITE ( 7,480 )DS
0391         END IF
0392     470 FORMAT ( A,'SINGLE PLATE' )
0393     480 FORMAT ( A,'DOUBLE PLATE' )
0394 C
0395         IF ( CTYPE.EQ.1 ) GO TO 520
0396 C
0397         IF ( PFUNC.EQ.1 ) THEN
0398             WRITE ( 7,485) DS
0399         ELSE IF ( PFUNC.EQ.2 ) THEN
0400             WRITE ( 7,486) DS
0401         ELSE IF ( PFUNC.EQ.3 ) THEN
0402             WRITE ( 7,487) DS
0403         END IF
0404     485 FORMAT ( A,'ORIGINAL PENETRATION FUNCTION' )
0405     486 FORMAT ( A,'PEN4 PENETRATION FUNCTION' )
0406     487 FORMAT ( A,'REGRESSION PENETRATION FUNCTION' )
0407 C
0408         WRITE ( 7,500 )DS,MATERIAL(MAT(1))
0409     500 FORMAT(A,'SHIELD MATERIAL = ',A)
0410 C
0411         WRITE ( 7,510 )DS,LUNITS,SHTHK
0412     510 FORMAT(A,'SHIELD THICKNESS (' ,A,') = ',F8.4)
0413 C
0414     520 WRITE ( 7,530 )DS,MATERIAL(MAT(2))
0415     530 FORMAT(A,'VESSEL WALL MATERIAL = ',A)
0416 C
0417         WRITE ( 7,540 )DS,LUNITS,VWTHK
0418     540 FORMAT(A,'VESSEL WALL THICKNESS (' ,A,') = ',F8.4)
0419 C
0420         IF ( CTYPE.EQ.1 ) GO TO 570
0421 C
0422         WRITE ( 7,550 )DS,LUNITS,STAND
0423     550 FORMAT(A,'SHIELD STAND-OFF (' ,A,') = ',F8.4)
0424 C
0425         IF ( MLI ) THEN
0426             WRITE ( 7,560 )DS
0427     560     FORMAT(A,'30 LAYERS OF MLI BETWEEN SHIELD AND VESSEL WALL')
0428         END IF
0429 C
0430 C   If the variables were read in in metric units, convert to english.
0431 C
0432     570 IF ( METRIC ) THEN
0433         SHTHK=SHTHK/2.54
0434         VWTHK=VWTHK/2.54
0435         STAND=STAND/2.54
0436     END IF
0437 C
0438 C   Build the material properties arrays.
0439 C
0440     DO 580 I=1,2
0441 C

```

```
0442          BHARD(I)=HARD(MAT(I))
0443          C(I)=SOUND(MAT(I))
0444          DENS(I)=DENSITY(MAT(I))
0445          FSU(I)=SHEAR(MAT(I))
0446          FTU(I)=ULT(MAT(I))
0447          FY(I)=YIELD(MAT(I))
0448          SHPV(I)=SHOCK(MAT(I))
0449          WILKC(I)=CONSTANT(MAT(I))
0450      C
0451      580 CONTINUE
0452      C
0453          RETURN
0454      C
0455          END
```

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE DOUBLE (ANGR,DIA,INITIAL,ITYPE,MLI,PFUNC,SHTHK,STAND,
0005 VELE,VELM,VWTHK,BHARD,C,DENS,FSU,FTU,FY,
0006 SHPV,WILKC)
0007 C
0008 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0009 C
0010 C
0011 C DOUBLE DETERMINES THE PROJECTILE DIAMETER THAT JUST PENETRATES
0012 C THE GIVEN DOUBLE PLATE CONFIGURATION AT THE GIVEN IMPACT VELOCITY
0013 C AND ANGLE. IT IS USED FOR BOTH THE ORIGINAL AND REGRESSION
0014 C PENETRATION FUNCTIONS.
0015 C
0016 C
0017 C
0018 C VARIABLE LIST
0019 C
0020 C ANGR = IMPACT ANGLE MEASURED FROM THE NORMAL,RADIANS
0021 C BALL = LOGICAL PARAMETER USED TO DETERMINE IF THE BALLISTIC
0022 C SUBROUTINES ARE CALLED
0023 C BHARD = ARRAY CONTAINING THE VALUES OF THE BRINNEL HARDNESS FOR
0024 C THE SHIELD & VESSEL WALL MATERIALS
0025 C BRIST = LOGICAL PARAMETER USED TO DETERMINE IF THE FRAGMENTING
0026 C SUBROUTINES ARE CALLED
0027 C C = ARRAY CONTAINING VALUES FOR THE SPEED OF SOUND (FT/SEC) FOR
0028 C THE SHIELD AND VESSEL WALL MATERIALS
0029 C DIA = PROJECTILE DIAMETER ( IN. )
0030 C DIAB = DIAMETER AS DETERMINED BY SUBROUTINE BRISTOW ( IN. )
0031 C DIABL = DIAMETER AS DETERMINED BY SUBROUTINE BALLIST ( IN. )
0032 C DIAW = DIAMETER AS DETERMINED BY SUBROUTINE WILKIN ( IN. )
0033 C EVWTHK = EQUIVALENT VESSEL WALL THK. ( IN. )
0034 C FY = ARRAY CONTAINING THE VALUES OF THE YIELD STRENGTH FOR THE
0035 C SHIELD & VESSEL WALL MATERIALS
0036 C FSU = ARRAY CONTAINING VALUES OF THE ULTIMATE SHEAR STRENGTH (PSI)
0037 C FOR THE SHIELD AND VESSEL WALL MATERIALS
0038 C FTU = ARRAY CONTAINING VALUES OF THE ULTIMATE TENSILE STRENGTH (PSI)
0039 C FOR THE SHIELD AND VESSEL WALL MATERIALS
0040 C INITIAL = LOGICAL PARAMETER USED TO DETERMINE IF CURRENT
0041 C SUBROUTINE CALL IS INITIAL ONE
0042 C ITYPE = ANALYSIS TYPE 1=DEBRIS & METEOROIDS, 2=METEOROIDS
0043 C MLI = CHARACTER STRING USED TO DETERMINE IF MLI IS USED IN WALL
0044 C PFUNC = PENETRATION FUNCTION
0045 C 1- ORIGINAL
0046 C 2- PEN4
0047 C 3- REGRESSION
0048 C SHATTER = LOGICAL PARAMETER USED TO DETERMINE IF PEN4 BALLISTIC
0049 C LIMITS HAVE BEEN EXCEEDED
0050 C SHPV = ARRAY CONTAINING THE VALUES FOR THE SHOCK PROJECTILE
0051 C VELOCITY (FT/SEC ) OF THE SHIELD & VESSEL WALL MATERIALS
0052 C SHTHK = SHIELD THICKNESS ( IN. )
0053 C STAND = SHIELD STAND-OFF DISTANCE (IN.)
0054 C THKMLI = EQUIVALENT THK OF 30 LAYERS OF MLI ( IN. )
0055 C VELE = VEL IN FT/SEC
0056 C VELM = VEL IN KM/SEC
0057 C VWTHK = VESSEL WALL THICKNESS ( IN. )
0058 C WILKC = ARRAY CONTAINING THE VALUES OF WILKINSON'S CONSTANT FOR
0059 C THE SHIELD & VESSEL WALL MATERIALS
0060 C
0061 C INTEGER*2 ITYPE,PFUNC
0062 C
0063 C DIMENSION BHARD(3),C(3),DENS(3),FY(3),FSU(3),FTU(3),SHPV(3),

```

```

0064          WILKC(3)
0065 C
0066     LOGICAL BALL,BRIST,INITIAL,MLI,SHATTER
0067 C
0068 C  INITIALIZE VARIABLES
0069 C
0070     IF ( INITIAL ) THEN
0071         BALL=.TRUE.
0072         BRIST=.TRUE.
0073         DIA=.01
0074         SHATTER=.FALSE.
0075     END IF
0076 C
0077 C  TAKE 30 LAYERS OF MLI INTO ACCOUNT USING COUR-PALAIS'S EQN.
0078 C  SET LIMIT FOR EQN. AT 10 KM/SEC OR SKIP OVER SECTION ENTIRELY
0079 C
0080 C
0081     IF (.NOT. MLI ) THEN
0082         THKMMLI = 0.0
0083         GO TO 50
0084     ENDIF
0085 C
0086     IF ( VELM .LT. 10.0 ) THEN
0087         THKMMLI = 3.045E-06 * ( VELM ** 3.42 )
0088     ELSE
0089         THKMMLI = 3.045E-06 * ( 10.0 ** 3.42 )
0090     END IF
0091 C
0092 C  CONVERT TO IN.
0093 C
0094     THKMMLI = THKMMLI / 2.54
0095 C
0096 C  ADD EQUIVALENT THK. OF MLI TO VESSEL WALL THK.
0097 C
0098 50     EVWTHK = VWTHK + THKMMLI
0099 C
0100 C
0101 C  DETERMINE DIAMETER IN IN. THAT PENETRATES THE SHIELD AND
0102 C  JUST DOES NOT PENETRATE THE VESSEL WALL
0103 C
0104 C  IF THE ANALYSIS IS FOR METEORIDS, THEN ONLY USE WILKINSON'S METHOD
0105 C  TO DETERMINE THE PENETRATION DIAMETER
0106 C
0107 C
0108     IF ( ITYPE .EQ. 2 .OR. VELM .GT. 12.0 ) THEN
0109         BALL=.FALSE.
0110         BRIST=.FALSE.
0111         GOTO 500
0112     ENDIF
0113 C
0114 C  INITIALLY CALCULATE THE PENETRATION DIAMETER USING BOTH THE
0115 C  BALLISTIC AND FRAGMENTING SUBROUTINES. THE DIAMETER CALCULATED BY
0116 C  THE BALLISTIC SUBROUTINE IS USED UNTIL THE VALUE CALCULATED FROM
0117 C  THE FRAGMENTING SUBROUTINE IS GREATER. AT THAT TIME IT IS NO LONGER
0118 C  NECESSARY TO CALL THE BALLISTIC SUBROUTINES.
0119 C
0120 C  FOR THE ORGINAL PENETRATION FUNCTION THE OLD VERSION OF PEN4 IS USED
0121 C  IN THE BALLISTIC REGIME AND THE BURCH MODIFIED EQUATIONS ARE USED
0122 C  IN THE FRAGMENTING REGIME.
0123 C
0124 C  FOR THE REGRESSION PENETRATION FUNCTION THE NEW PEN4 IS USED IN
0125 C  THE BALLISTIC REGIME AND THE REGRESSION EQUATIONS ARE USED IN
0126 C  FRAGMENTING REGIME.

```

```

0127 C
0128 C
0129 IF (BALL) THEN
0130 IF ( PFUNC.EQ.1 ) THEN
0131 CALL BALLIST (ANGR,BHARD,C,DENS,DIA,VWTHK,FY,FSU,FTU,
0132 1 INITIAL,SHATTER,SHPV,SHTHK,STAND,VELE)
0133 ELSE
0134 CALL PEN4 ( ANGR,BHARD,C,DENS,DIA,VWTHK,FY,FSU,FTU,
0135 1 INITIAL,SHATTER,SHPV,SHTHK,STAND,VELE )
0136 END IF
0137 DIABL=DIA
0138 C
0139 IF ( PFUNC.EQ.1. ) THEN
0140 CALL BRISTOW (ANGR,C,DIA,EVWTHK,INITIAL,PFUNC,SHTHK,STAND,
0141 1 VELE )
0142 ELSE
0143 CALL REGRESS ( ANGR,DIA,MLI,SHTHK,STAND,VELM,VWTHK )
0144 END IF
0145 DIAB=DIA
0146 C
0147 C CHECK IF THE DIAMETER CALCULATED BY BALLISTIC SUBROUTINE IS LESS
0148 C THAN THAT CALCULATED BY FRAGMENTING SUBROUTINE. IF SO SET BALL TO
0149 C FALSE .
0150 C
0151 IF (DIAB.GT.DIABL) THEN
0152 BALL=.FALSE.
0153 DIA=DIAB
0154 GOTO 700
0155 ELSE
0156 BALL=.TRUE.
0157 DIA=DIABL
0158 GOTO 700
0159 ENDIF
0160 C
0161 C CALCULATE THE PENETRATION DIAMETER USING BOTH THE FRAGMENTING AND
0162 C WILKIN SUBROUTINES. THE DIAMETER CALCULATED BY FRAGMENTING IS USED
0163 C UNTIL THE VALUE DETERMINED BY WILKIN IS LESS. IT IS THEN NOT
0164 C NECESSARY TO CALL FRAGMENTING.
0165 C
0166 C
0167 ELSE
0168 IF (BRIST) THEN
0169 IF ( PFUNC.EQ.1. ) THEN
0170 CALL BRISTOW (ANGR,C,DIA,EVWTHK,INITIAL,PFUNC,SHTHK,STAND,
0171 1 VELE )
0172 ELSE
0173 CALL REGRESS ( ANGR,DIA,MLI,SHTHK,STAND,VELM,VWTHK )
0174 END IF
0175 DIAB=DIA
0176 C
0177 CALL WILKIN (ANGR,DENS,DIA,EVWTHK,ITYPE,SHTHK,STAND,VELM,
0178 WILKC)
0179 DIAW=DIA
0180 C
0181 C CHECK IF THE VALUE DETERMINED BY WILKIN IS LESS THEN THAT
0182 C DETERMINED BY BRISTOW. IF SO SET BRIST TO FALSE.
0183 C
0184 IF (DIAW.LT.DIAB) THEN
0185 BRIST=.FALSE.
0186 DIA=DIAW
0187 GOTO 700
0188 ELSE
0189 BRIST=.TRUE.

```

D180-30550-4

```

0190          DIA=DIAB
0191          GOTO 700
0192      ENDIF
0193  C
0194      ENDIF
0195  C
0196      ENDIF
0197  C
0198  C
0199  C  CALCULATE THE DIAMETER USING THE WILKIN SUBROUTINE.
0200  C
0201  500  CALL WILKIN (ANGR,DENS,DIA,EVWTHK,ITYPE,SHTHK,STAND,VELM,WILKC)
0202  C
0203  C  SET INITIAL TO FALSE
0204  C
0205  700  INITIAL = .FALSE.
0206  C
0207  C
0208      RETURN
0209  C
0210      END

```

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE BRISTOW (ANGR,C,DIA,EVWTHK,INITIAL,SHTHK,STAND,
0005 1 VELE )
0006 C
0007 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008 C
0009 C
0010 C
0011 C BRISTOW DETERMINES NUMBER OF PLATES PENETRATED. BASED ON
0012 C "MULTIPLE-DAMAGE STUDY", BY G.T. BURCH, BOEING 1967,AIR FORCE
0013 C ARMAMENT LABORATORY TECH REPORT AFATL-TR-67-116.
0014 C
0015 C VALID FROM 3.6 TO 10.2 KM/SEC,AND FOR ALUMINUM PROJECTILES
0016 C IMPACTING ALUMINUM PLATES AT NORMAL .
0017 C
0018 C
0019 C VARIABLE LIST
0020 C
0021 C ANGR = IMPACT ANGLE MEASURED FROM THE NORMAL, RADIANS
0022 C C1 = SPEED OF SOUND IN SHIELD MATERIAL,FT/SEC
0023 C C = ARRAY CONTAINING VALUES OF THE SPEED OF SOUND FOR
0024 C THE SHIELD AND VESSEL WALL MATERIALS ( FT/SEC )
0025 C CHI = INTERMEDIATE VARIABLE
0026 C DIA = PROJECTILE DIAMETER,IN.
0027 C DIA1 = PREVIOUS DIAMETER
0028 C DIA2 = CURRENT DIAMETER
0029 C DT = INTERMEDIATE VARIABLE
0030 C EVWTHK = EQUIVALENT VESSEL WALL THICKNESS,IN.
0031 C F1 = INTERMEDIATE VARIABLE
0032 C F2 = " "
0033 C F3 = " "
0034 C I = ITERATION COUNTER
0035 C INITIAL = LOGICAL PARAMETER USED TO DETERMINE IF CURRENT CALL
0036 C IS INITIAL ONE
0037 C NF = NUMBER OF PLATES PENETRATED BY THE FLIGHT PATH COMPONENT
0038 C NN = NUMBER OF PLATES PENETRATED BY THE NORMAL COMPONENT
0039 C PLP = # OF PLATES PENETRATED
0040 C PLP1 = PREVIOUS PLP
0041 C PLP2 = CURRENT PLP
0042 C SD = INTERMEDIATE VARIABLE
0043 C SHTHK = SHIELD THICKNESS,IN.
0044 C SIN3 = INTERMEDIATE VARIABLE
0045 C SLOPE = SLOPE OF LINE THROUGH (DIA1,PLP1) & (DIA2,PLP2)
0046 C SPF = NUMBER OF PLATES PENETRATED THAT ACCOUNTS FOR SPALLING
0047 C STAND = SHIELD STAND-OFF (IN.)
0048 C SWITCH = LOGICAL VARIABLE USED TO DETERMINE IF THE CRITICAL
0049 C DIAMETER IS TO BE ESTIMATED THROUGH LINEAR
0050 C INTERPOLATION
0051 C T1D = INTERMEDIATE VARIABLE
0052 C T2D = "
0053 C TEST = RATIO OF THE NUMBER OF PLATES PENETRATED TO THE
0054 C SPALLING FACTOR
0055 C VC = "
0056 C VELE = VEL IN FT/SEC
0057 C VHOLD = VELOCITY HOLDER
0058 C
0059 C
0060 C
0061 C DIMENSION C(3)
0062 C
0063 C LOGICAL INITIAL

```

```

0064 C
0065 REAL NF,NN
0066 C
0067 LOGICAL SWITCH
0068 C
0069 SAVE PLP1,PLP2,DIA1,DIA2
0070 C
0071 C
0072 C SINCE THIS METHOD DOES NOT SOLVE FOR THE DIAMETER THAT JUST
0073 C PENETRATES DIRECTLY AN ITERATIVE APPROACH IS TAKEN. INITIAL VALUES
0074 C ARE SET,AND USED TO APPROXIMATE THE CORRECT VALUE. THIS PROCESS
0075 C IS CONTINUED UNTIL THE ANSWER IS WITHIN TOLERANCES.
0076 C
0077 C
0078 C SET INITIAL VALUES
0079 C
0080 I=0
0081 SPF = 0.85
0082 SWITCH=.FALSE.
0083 C
0084 C IF THIS IS THE INITIAL CALL SET INITIAL ELSE USE PREVIOUS VALUES
0085 C VALUES
0086 C
0087 IF (INITIAL) THEN
0088 DIA2=10.0
0089 ENDIF
0090 C
0091 C FOR VELOCITIES LESS THAN 11,800 FT/SEC SET THE VELOCITY EQUAL
0092 C TO 11,800 AND CALCULATE THE CRITICAL DIAMETER. USE THIS VALUE
0093 C TO ESTIMATE THE ACTUAL DIAMETER. SET THE LOGICAL VARIABLE SWITCH
0094 C TO TRUE AND SAVE THE VELOCITY AS VHOLD.
0095 C
0096 IF ( VELE .LT. 11800. ) THEN
0097 VHOLD=VELE
0098 VELE=11800.
0099 SWITCH=.TRUE.
0100 END IF
0101 C
0102 C CALCULATE INTERMEDIATE VARIABLES THAT DO NOT NEED TO BE CALCULATED
0103 C FOR EACH DIAMETER
0104 C
0105 CHI=TAN(ANGR)-0.50
0106 SIN3=(SIN(ANGR))**3
0107 VC=VELE/C(1)
0108 C
0109 100 I=I+1
0110 C
0111 C IF THIS IS THE FIRST PREDICTION USE THE DIAMETER VALUE THAT WAS
0112 C CALCULATED TO PENETRATE FOR THE PREVIOUS CASE AS A STARTING POINT.
0113 C ELSE USE A LINEAR PREDICTION APPROACH BASED ON THE LAST TWO
0114 C PREDICTIONS.
0115 C
0116 IF ( I.EQ.1 ) THEN
0117 C
0118 DIA=DIA2
0119 DIA2=0.0
0120 PLP2=0.0
0121 C
0122 ELSE
0123 C
0124 SLOPE=(PLP2-PLP1)/(DIA2-DIA1)
0125 DIA=((SPF-PLP1)/SLOPE+DIA1)
0126 C

```

D180-30550-4

```

0127 C CHECK THAT DIA > 0.0
0128 C
0129 IF (DIA.LT.0.0) DIA=1.0E-10
0130 C
0131 C Check if dial = dia2,if so stop
0132 C
0133 IF ( DIA.EQ.DIA2 ) THEN
0134 WRITE ( 6,150 ) ANGR,VELE,DIA
0135 150 FORMAT (/1X,'BRISTOW CANNOT CONVERGE BECAUSE OF FLAT ',
0136 1 'SLOPE (ANGLE,VEL,DIA) =',3E12.5 )
0137 STOP
0138 END IF
0139 END IF
0140 C
0141 C CALCULATE # OF PLATES PENETRATED
0142 C
0143 SD=STAND/DIA
0144 T1D=SHTHK/DIA
0145 T2D=EVWTHK/DIA
0146 C
0147 F1=2.42*(T1D**(-1./3.))+4.26*(T1D**(1./3.))-4.18
0148 C
0149 F2=(0.5-1.87*T1D)+(5.*T1D-1.6)*(CHI**3)+(1.7-12.*T1D)*CHI
0150 C
0151 F3=0.32*(T1D**(5./6.))+0.48*(T1D**(1./3.))*SIN3
0152 C
0153 NF=(F1+0.63*F2)*(VC**(-4./3.))*(SD**(-5./12.))*(T2D**(-7./12.))
0154 C
0155 NN=F3*(T2D**(-1.))*(VC**(-4./3.))
0156 C
0157 C DETERMINE WHICH COMPONENT CONTROLS
0158 C
0159 IF ( NF.GT.NN ) THEN
0160 PLP=NF
0161 ELSE
0162 PLP=NN
0163 ENDIF
0164 C
0165 C RESET HOLDERS
0166 C
0167 DIA1=DIA2
0168 DIA2=DIA
0169 PLP1=PLP2
0170 PLP2=PLP
0171 C
0172 C CHECK IF PLP IS WITHIN TOLERANCE,IF NOT ITERATE
0173 C
0174 TEST = PLP/SPF
0175 C
0176 C
0177 IF ( TEST.LT.0.99 .OR. TEST.GT.1.01 ) GO TO 100
0178 C
0179 C IF SWITCH IS TRUE, ESTIMATE THE CRITICAL DIAMETER USING LINEAR
0180 C INTERPOLATION. THE TWO POINTS USED ARE THE ORIGIN AND THE VALUE
0181 C CALCULATED AT 11800 .
0182 C
0183 IF ( SWITCH ) THEN
0184 C
0185 DIA=DIA2*VHOLD/11800.0
0186 SWITCH=.FALSE.
0187 DIA1=0.0
0188 PLP1=0.0
0189 C

```

D180-30550-4

```
0190      END IF
0191      C
0192      RETURN
0193      C
0194      C
0195      END
```

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE WILKIN (ANGR,DENS,DIA,EVWTHK,ITYPE,SHTHK,STAND,VELM,
0005 WILKC )
0006 C
0007 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008 C
0009 C WILKIN DETERMINES THE DIAMETER THAT JUST PENETRATES THE VESSEL
0010 C WALL. IT IS BASED ON J.P.D. WILKINSON'S PAPER 'A PENETRATION
0011 C CRITERION FOR DOUBLED-WALLED STRUCTURE SUBJECT TO METEOROID IMPACT'
0012 C ,AIAA JOURNAL OCT 1969.
0013 C
0014 C THE MAJOR ASSUMPTION USED BY WILKINSON IS THAT THE PROJECTILE
0015 C VAPORIZES ON IMPACT
0016 C
0017 C
0018 C VARIABLE LIST
0019 C
0020 C ANGR = IMPACT ANGLE MEASURED FROM THE NORMAL, RADIANS
0021 C CONST = MATERIAL CONSTANT DEFINED BY WILKINSON
0022 C C1 = INTERMEDIATE VARIABLE
0023 C DENS = ARRAY CONTAINING VALUES FOR DENSITY OF THE SHIELD &
0024 C VESSEL WALL MATERIALS
0025 C DIA = PROJECTILE DIAMETER (IN)
0026 C DIAM = PROJECTILE DIAMETER (CM )
0027 C EVWTHK = EQUIVALENT VESSEL WALL THK. ( IN)
0028 C ITYPE = ANALYSIS TYPE 1=DEBRIS & METEORIDS, 2=METEORIDS
0029 C MASS = PROJECTILE MASS ( GRAMS )
0030 C MB = INTERMEDIATE VARIABLE
0031 C MT = " "
0032 C PI = PI
0033 C PROJDEN = PROJECTILE DENSITY ( G/CC )
0034 C RMI = UNIT MASS OF SHIELD (G/CM**2)
0035 C RM2 = UNIT MASS OF VESSEL WALL (G/CM**2)
0036 C SHDEN = SHIELD & VESSEL WALL DENISITY ( GRAMS/CC )
0037 C SHTHK = SHIELD THK. (IN)
0038 C STAND = SHIELD STAND-OFF DISTANCE (IN)
0039 C STANDM = STAND IN CM.
0040 C VELM = VEL IN KM/SEC
0041 C VNORM = NORMAL COMPONENT OF THE VELEOCITY VECTOR,KM/SEC
0042 C WVDEN = VESSEL WALL DENSITY (LB/IN**3)
0043 C WILKC = ARRAY CONTAINING VALUES OF WILKINSON'S CONSTANT
0044 C FOR THE SHIELD & VESSEL WALL MATERIALS
0045 C
0046 C
0047 C
0048 DIMENSION DENS(3),WILKC(3)
0049 C
0050 INTEGER*2 ITYPE
0051 C
0052 REAL MB,MT,MASS
0053 C
0054 C SET INITIAL VALUES
0055 C
0056 PI = 3.141592654
0057 C
0058 C SET PROJECTILE DENSITY IN G/CC
0059 C
0060 PROJDEN = DENS(3)*27.705
0061 C
0062 C SET SHIELD AND VESSEL WALL DENSITY IN G/CC
0063 C

```

D180-30550-4

```

0064      SHDEN = DENS(1) * 27.705
0065      VWDEN = DENS(2) * 27.705
0066      C
0067      C  CONST IS A MATERIAL VARIABLE DEFINED IN THE PAPER & IS
0068      C  EQUAL TO .401 FOR 2219 AL
0069      C
0070      CONST = WILKC(2)
0071      C
0072      C  DETERMINE SHIELD & VESSEL WALL MASS PER UNIT AREA
0073      C
0074      RM1 = SHTHK * 2.54 * SHDEN
0075      RM2 = EVWTHK * 2.54 * VWDEN
0076      C
0077      C  CONVERT STAND TO CM
0078      C
0079      STANDM = STAND * 2.54
0080      C
0081      C  CALCULATE THE NORMAL COMPONENT OF THE VELOCITY VECTOR
0082      C
0083      VNORM=VELM*COS(ANGR)
0084      C
0085      C  DETERMINE CRITICAL PROJECTILE DIAMETER
0086      C
0087      MT = 1.44*(PI/6.0)**(1./3.)*CONST*RM1*RM2*STANDM**2.0
0088      MB = PROJDN**(2./3.)*VNORM
0089      MASS = (MT/MB)**.75
0090      DIAM = (6.0*MASS/(PI*PROJDN))**(1./3.)
0091      C
0092      C  CHECK IF APPROPRIATE EQ. WAS USED
0093      C
0094      C1 = RM1 / ( PROJDN * DIAM )
0095      IF ( C1 .GT. 1.0 ) THEN
0096      C
0097      C  WRONG EQN. USED, RECALC USING CORRECT EQN.
0098      C
0099      MASS= 1.44*CONST*RM2*(STANDM**2.)/VNORM
0100      DIAM = (6.0*MASS/(PI*PROJDN))**(1./3.)
0101      END IF
0102      C
0103      C  CONVERT DIAMETER TO IN
0104      C
0105      DIA = DIAM / 2.54
0106      C
0107      RETURN
0108      C
0109      END

```

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE REGRESS ( ANGR,DIA,MLI,SHTHK,STAND,VELM,VWTHK )
0005 C
0006 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007 C
0008 C
0009 C
0010 C Subroutine REGRESS determines the critical projectile diameter for
0011 C a two-plate structure impacted by aluminum spherical projectiles. It
0012 C was developed from test data obtained during the NASA contract
0013 C 'Integrated Space Station Wall Design Guide and Penetration Control
0014 C Plan'. The data varied from 2 to 8 km/sec.
0015 C
0016 C
0017 C Variable List
0018 C
0019 C angr = impact angle measured from the normal, radians
0020 C dia = critical projectile diameter, in
0021 C dl3 = intermediate variable
0022 C high = holder, last diameter to penetrate , in
0023 C ic = counter
0024 C low = holder, last diameter to not penetrate, in
0025 C lsd = intermediate variable
0026 C mli = logical variable indicating wether multi-layer insulation is
0027 C used
0028 C mt = intermediate variable
0029 C np = number of plates penetrated, excluding the shield
0030 C shthk = shield thickness , in
0031 C stand = shield stand-off , in
0032 C switch = logical variable, used to determine if a penetrating
0033 C diameter has been determined
0034 C tt = intermediate variable
0035 C t13 = " "
0036 C vc2 = " "
0037 C velm = impact velocity, km/sec
0038 C vwthk = vessel wall thickness , in
0039 C
0040 C
0041 C
0042 C INTEGER IC
0043 C
0044 C REAL LOW,LSD,MT,NP
0045 C
0046 C LOGICAL MLI,SWITCH
0047 C
0048 C Intialize Variables
0049 C
0050 C IC=0
0051 C LOW=0.0
0052 C HIGH=5.0
0053 C SWITCH=.TRUE.
0054 C
0055 C Set MLI constant
0056 C
0057 C IF ( MLI ) THEN
0058 C MT=-14.0
0059 C ELSE
0060 C MT=0.0
0061 C END IF
0062 C
0063 C Since the equation does not solve for the critical projectile

```

D180-30550-4

```

0064 C diameter directly, use a binary search technique to determine it.
0065 C First determine a diameter that penetrates then narrow in on the
0066 C actual diameter.
0067 C
0068 100 CONTINUE
0069 C
0070 IC=IC+1
0071 C
0072 IF ( SWITCH ) THEN
0073     HIGH=HIGH*2.0
0074 END IF
0075 C
0076 DIA=(HIGH+LOW)/2.0
0077 C
0078 C Check that the diameter is less then 50 cm , if not stop
0079 C
0080 IF ( DIA.GT.50.0 ) THEN
0081     WRITE ( 6,(''---ERROR--- Diameter greater than 50 cm in '',
0082 1     ''subroutine REGRESS''))
0083     STOP
0084 END IF
0085 C
0086 C Calculate the intermediate variables
0087 C
0088 D13=DIA**(1./3.)
0089 LSD=LOG10(STAND)/DIA
0090 T13=SHTHK**(1./3.)
0091 TT=TAN(ANGR)
0092 VC2=VELM*(COS(ANGR)**2)
0093 C
0094 C Calculate the number of plates penetrated
0095 C
0096 NP=1.52-6.18*T13-18.8*VWTHK-0.146*LSD+MT*SHTHK+10.8*D13-0.287*VC2
0097 1 -0.713*TT
0098 C
0099 C Check for convergence
0100 C
0101 IF ( IC.GT.100 ) THEN
0102     WRITE ( 6,(''---ERROR--- REGRESS failed to converge after '',
0103 1     ''100 cycles'' )')
0104     STOP
0105 END IF
0106 C
0107 C Has a diameter that penetrates been found, if not reset holders and
0108 C try again
0109 C
0110 IF ( SWITCH ) THEN
0111     IF ( NP.GT.1 ) THEN
0112         SWITCH=.FALSE.
0113         HIGH=DIA
0114         GO TO 100
0115     END IF
0116 END IF
0117 C
0118 C Does the diameter yeild an acceptable result, if not rest holders
0119 C and try again
0120 C
0121 IF ( NP.LT.0.999 ) THEN
0122     LOW=DIA
0123     GO TO 100
0124 ELSE IF ( NP.GT.1.001 ) THEN
0125     HIGH=DIA
0126     GO TO 100

```

0127		END IF
0128	C	
0129	C	Finished
0130	C	
0131		RETURN
0132	C	
0133		END

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE BALLIST (ANGR,BHARD,C,DENS,DIA,VWTHK,FY,FSU,FTU,
0005 1 INITIAL,SHATTER,SHPV,SHTHK,STAND,VELE )
0006 C
0007 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008 C
0009 C BALLISTIC DETERMINES THE DIAMETER THAT JUST PENETRATES. IT UTILIZES
0010 C THE BALLISTIC PORTION OF BOEING'S HYPERVELOCITY CODE PEN4.
0011 C
0012 C
0013 C VARIABLE LIST
0014 C
0015 C ANGR = IMPACT ANGLE MEASURED FROM THE NORMAL,RADIANS
0016 C BHARD = ARRAY CONTAINING THE VALUES OF THE BRINNEL HARDNESS FOR
0017 C THE SHIELD & VESSEL WALL MATERIALS
0018 C C = ARRAY CONTAINING VALUES FOR THE SPEED OF SOUND FOR THE
0019 C SHIELD AND VESSEL WALL MATERIALS (FT/SEC)
0020 C DEN = PROJECTILE DENSITY (LB/CFT)
0021 C DENS = ARRAY CONTAINING VALUES FOR THE DENSITY FOR THE
0022 C SHIELD AND VESSEL WALL MATERIALS (LB/IN**3)
0023 C DIA = SPHERICAL PROJECTILE DIAMETER (IN)
0024 C DIA1 = PREVIOUS DIAMETER THAT PENETRATED (IN)
0025 C DIA2 = DIAMETER THAT PENETRATED TWO ITERATIONS PREVIOUSLY (IN)
0026 C EVWTHK = EQUIVALENT VESSEL WALL THICKNESS,IN.
0027 C FY = ARRAY CONTAINING THE VALUES OF THE YIELD STRENGTH FOR THE
0028 C SHIELD & VESSEL WALL MATERIALS
0029 C FSU = ARRAY CONTAINING VALUES OF THE ULTIMATE SHEAR STRENGTH
0030 C FOR THE SHIELD AND VESSEL WALL MATERIALS (PSI)
0031 C FTU = ARRAY CONTAINING VALUES OF THE ULTIMATE TENSILE STRENGTH
0032 C FOR THE SHIELD AND VESSEL WALL MATERIALS (PSI)
0033 C I = COUNTER
0034 C INITIAL = LOGICAL PARAMETER USED TO DETRMINIE IF CURRENT CALL
0035 C IS INITIAL ONE.
0036 C MASS = PROJECTILE MASS,LBS
0037 C N = INCREMENT MULIPLIER
0038 C PDENS = DENS ARRAY CONVERTED TO SLUGS/FT**3
0039 C PFY = FY ARRAY CONVERTED TO PSF
0040 C PFSU = FSU ARRAY CONVERTED TO PSF
0041 C PFTU = FTU ARRAY CONVERTED TO PSF
0042 C P1 = LAST MASS GUESS TO NOT PENETRATE
0043 C P2 = LAST MASS GUESS TO PENETRATE
0044 C PEN = TRUE OR FALSE
0045 C PI = 3.14
0046 C PMINCR = INITIAL MASS GUESS INCREMENT
0047 C RATIO = P2 / P1
0048 C SHATTER = LOGICAL PARAMETER USED TO DETERMINE IF PEN4 BALLISTIC
0049 C LIMITS HAVE BEEN EXCEEDED
0050 C SHPV = ARRAY CONTAING VALUES OF THE SHOCK PROJECTILE VELOCITY
0051 C FOR THE SHIELD AND VESSEL WALL MATERIALS,( UNITLESS )
0052 C SHTHK = SHIELD THICKNESS,IN.
0053 C SPACE = ARRAY CONTAINING THE SHIELD SPACING ,FT.
0054 C STAND = SHIELD STAND-OFF,IN.
0055 C TARMAT = ARRAY CONTAINING MATERIAL PROPERTY POINTERS
0056 C THETA = IMPACT ANGLE (RAD),MEASURED FROM THE NORMAL
0057 C THICK = SHIELD & VESSEL WALL THICKNESS,FT.
0058 C VELE = COLLISION VELOCITY,FT/SEC
0059 C VEL1 = VEL FOR DIA1
0060 C VEL2 = VEL FOR DIA2
0061 C V0 = IMPACT VELOCITY (FT/SEC)
0062 C VR2 = RESIDUAL VELOCITY AFTER VESSEL WALL,FT/SEC
0063 C

```

```

0064 C
0065 C
0066 DIMENSION BHARD(3),C(3),DENS(3),FY(3),FSU(3),FTU(3),PDENS(3),
0067 . PFY(3),PFSU(3),PFTU(3),SHPV(3),SPACE(2),THICK(3),
0068 . TARMAT(3)
0069 C
0070 INTEGER PROJMAT,TARMAT
0071 C
0072 LOGICAL PEN,INITIAL,SHATTER
0073 C
0074 REAL MASS
0075 C
0076 SAVE DIA1,DIA2,P2,VEL1,VEL2
0077 C
0078 C
0079 C SET INITIAL VALUES
0080 C
0081 I = 0
0082 N = 0
0083 PEN = .FALSE.
0084 PMINCR = 1.0E-04
0085 . PI = 3.1415926536
0086 P1 = 1.0E-06
0087 C
0088 C CONVERT VARIABLES TO PEN4 FORMAT AND UNITS
0089 C
0090 DO 50 I=1,3
0091 C
0092 C CONVERT DENS TO SLUGS/FT**3
0093 C
0094 PDENS(I)=DENS(I)*1728./32.2
0095 C
0096 C CONVERT FSU AND FTU TO PSF
0097 C
0098 PFY(I)=FY(I)*144.
0099 PFTU(I)=FTU(I)*144.
0100 PFSU(I)=FSU(I)*144.
0101 C
0102 50 CONTINUE
0103 C
0104 PROJMAT = 3
0105 SPACE(1) = STAND/12.0
0106 SPACE(2) = 0.0
0107 TARMAT(1) = 1
0108 TARMAT(2) = 2
0109 TARMAT(3) = 0
0110 THETA = ANGR
0111 THICK(1) = SHTHK/12.0
0112 THICK(2) = VWTHK/12.0
0113 THICK(3) = 0.0
0114 V0 = VELE
0115 C
0116 C PEN4 DOES NOT SOLVE FOR THE CRITICAL PROJECTILE DIAMETER DIRECTLY
0117 C THEREFORE AN ITERATIVE APPROACH IS TAKEN. PEN4 ONLY RETURNS
0118 C A LOGICAL PARAMETER INDICATING IF PENETRATION HAS OCCURED.
0119 C
0120 C
0121 C FOR INITIAL CASE SET P2 > 0.0
0122 C
0123 IF (INITIAL) THEN
0124 P2 = 1.0E-04
0125 ENDIF
0126 C

```

D180-30550-4

```

0127 C IF SHATTER HAS OCCURED NO NEED TO CALL PEN4,USE LINEAR APPROX.
0128 C
0129 IF (SHATTER) GO TO 500
0130 C
0131 C DETERMINE INITIAL MASS THAT PENETRATES
0132 C
0133 100 MASS = P2 + N * PMINCR
0134 C
0135 NC = NC + 1
0136 C
0137 CALL OPEN4 (PEN,MASS,THETA,SHATTER,V0,PDENS,BHARD,PROJMAT,SHPV,
0138 C,SPACE,TARMAT,THICK,PFY,*500)
0139 C
0140 IF ( PEN ) THEN
0141 C
0142 C IF PENETRATION TRY WITH MASS BETWEEN P1 & P2
0143 C
0144 PEN = .FALSE.
0145 P2 = MASS
0146 MASS = ( P1 + P2 ) / 2.0
0147 C
0148 NC = NC + 1
0149 CALL OPEN4 (PEN,MASS,THETA,SHATTER,V0,PDENS,BHARD,PROJMAT,
0150 SHPV,C,SPACE,TARMAT,THICK,PFY,*500)
0151 C
0152 C
0153 200 IF ( PEN ) THEN
0154 C
0155 C IF PENETRATION,SET P2 = MASS & CHECK RATIO
0156 C
0157 PEN = .FALSE.
0158 P2 = MASS
0159 RATIO = P2 /P1
0160 IF ( RATIO .LT. .995 .OR. RATIO .GT. 1.005 ) THEN
0161 C
0162 C IF TRUE TRY AGAIN BETWEEN P1 & P2
0163 C
0164 MASS = ( P1 + P2 ) / 2.0
0165 NC = NC + 1
0166 CALL OPEN4 (PEN,MASS,THETA,SHATTER,V0,PDENS,BHARD,
0167 PROJMAT,SHPV,C,SPACE,TARMAT,THICK,PFY,
0168 *500)
0169 GO TO 200
0170 ELSE
0171 C
0172 C IF FALSE,MASS IS WITHIN RANGE SO CONTINUE
0173 C
0174 GO TO 300
0175 END IF
0176 C
0177 ELSE
0178 C
0179 C IF FALSE,SET P1 = MASS AND CHECK RATIO
0180 C
0181 P1 = MASS
0182 RATIO = P2 / P1
0183 IF ( RATIO .LT. .995 .OR. RATIO .GT. 1.005 ) THEN
0184 C
0185 C IF TRUE,TRY AGAIN BETWEEN P1 & P2
0186 C
0187 MASS = ( P1 + P2 ) / 2.0
0188 NC = NC + 1
0189 CALL OPEN4 (PEN,MASS,THETA,SHATTER,V0,PDENS,BHARD,

```

D180-30550-4

```

0190      .      PROJMAT, SHPV, C, SPACE, TARMAT, THICK, PFY,
0191      .      *500)
0192      GO TO 200
0193      ELSE
0194      C
0195      C      IF FALSE, MASS IS WITHIN RANGE SO CONTINUE
0196      C
0197      GO TO 300
0198      END IF
0199      ENDIF
0200      ELSE
0201      C
0202      C      IF FALSE INCREASE INCREMENT & TRY AGAIN
0203      C
0204      P1 = MASS
0205      I = I + 1
0206      N = 2 ** I
0207      GO TO 100
0208      END IF
0209      C
0210      P2=MASS
0211      C
0212      C      CALCULATE DIAMETER
0213      C
0214      C
0215      C
0216      300      DEN = DENS(3)*1728.0
0217      C
0218      DIA = ((MASS*6.0)/(PI*DEN))**(1.0/3.0)
0219      C
0220      C      CONVERT TO IN.
0221      C
0222      DIA = DIA * 12.0
0223      C
0224      C
0225      C
0226      C      RESET HOLDERS
0227      C
0228      DIA2=DIA1
0229      DIA1=DIA
0230      VEL2=VEL1
0231      VEL1=VELE
0232      C
0233      C
0234      RETURN
0235      C
0236      C      SINCE SHATTER HAS OCCURED NO NEED TO CALL PEN4 USE LINEAR APPROXL
0237      C
0238      500      SLOPE=(DIA1-DIA2)/(VEL1-VEL2)
0239      DIA=DIA2+SLOPE*(VELE-VEL2)
0240      RETURN
0241      C
0242      END

```

D180-30550-4

```

0001 C
0002 SUBROUTINE OPEN4 (Penetration,Mass,Theta,SHATTER,V0,
0003 1 Density,Hardness,ProjMat,ShockProjVel,SoundVel,
0004 2 Spacing,TarMat,Thick,YieldStrength,*)
0005 C This version of pen4 is strictly for use in the sub-shatter velocity
0006 C regime. It is a modified version of spin14, created on 6/21/85.
0007
0008 LOGICAL Penetration,SHATTER
0009 REAL Mass,LAMBDA,NR,NF,NH,J,NR2,NHT,LastPK,MR
0010 INTEGER ProjMat,TarMat,Exponent,TopCount,BottomCount1,BottomCount2
0011 DIMENSION Thick(3),TarMat(3),Spacing(2),
0012 : Density(3),YieldStrength(3),SoundVel(3),
0013 : ShockProjVel(3),Hardness(3),Epsil(2),
0014 : NR(3),NR2(3),RF(3),Flagit(3)
0015 C
0016 C Density in Slugs/CubicFoot
0017 C YieldStrength in Lbs/SquareFoot
0018 C SoundVel in Feet/Second
0019 C Hardness is Brinell Scale
0020 DATA Epsil,Gamma/5.71,5.71,90.0/
0021 DATA F1,F2/4.0,1.0/
0022 DATA EffectiveThicknessRatio/.6/
0023 DATA A,B/2.0,3.125E-04/
0024 PI=3.141592653589793
0025 Small = 1E-36
0026 RecipSqrt2PI=1./SQRT(2.*PI)
0027 C *****Calculate Radius of Projectile Sphere*****
0028 RP=(Mass*3./
0029 1 (Density(ProjMat)*32.2*4.*PI))**(1./3.)
0030 Diam = RP * 2.
0031 C *****
0032
0033 C *****Compute ResidualVel*****
0034 VDEL=( V0**2*1.33*RP**2*Density(ProjMat)-
0035 1 F1*YieldStrength(TarMat(1))*THICK(1)**2/COS(THETA)**2
0036 2 *A*EXP(-B*V0) )
0037 2 /( 1.33*RP**2*Density(ProjMat)+F2*RP*THICK(1)*
0038 3 Density(TarMat(1))/COS(THETA) )
0039 VR=SQRT(AMAX1(VDEL,0.))
0040 C *****
0041
0042 C *****Cratering V50*****
0043 V50=SQRT((Thick(1)*EffectiveThicknessRatio/COS(Theta)/
0044 1 (0.281*Diam*(Density(ProjMat)/Density(TarMat(1)))*(1./3.)))
0045 2 *(1./31)/Density(ProjMat)*(2.*YieldStrength(TarMat(1))))
0046 C *****
0047
0048 C *****IF No Penetration Report Result and Exit*****
0049 IF (V0.LT.V50) THEN
0050 Penetration = .FALSE.
0051 RETURN
0052 ELSE IF (Thick(2).EQ.0.0) THEN
0053 Penetration = .TRUE.
0054 RETURN
0055 ENDIF
0056 C *****
0057
0058 C *****Mass Loss Regime Decision*****
0059 ToverD = Thick(1)/Diam
0060 FTHETA=-2.433E-4 - 1.643E-2*COS(THETA)+3.201*COS(THETA)**2-
0061 : 2.184*COS(THETA)**3
0062 IF (ToverD.LT..40) THEN
0063 Vf = 4100.

```

```

0064      ELSE
0065          Vf = 4986.*ToverD**.21
0066      END IF
0067      IF (V0.LT.VF+4000.) THEN
0068          IF (V0.LT.Vf) THEN
0069              CALL MassErr(V0,Mass,Hardness(ProjMat),
0070 :              Density(ProjMat),Density(TarMat(1)),Thick(1),Diam,MR)
0071              ELSE
0072                  CALL Fract(MR,V0,Vf,FTheta,Mass,ToverD)
0073              END IF
0074              CALL LarMR(VR,MR,V50,Thick(2),EffectiveThickNessRatio,Theta,
0075 :              Diam,Density(ProjMat),Density(TarMat(2)),
0076 :              YieldStrength(TarMat(1)),Penetration)
0077      ELSE
0078      C          *****The rest of this subroutine contains the evaluation*****
0079      C          ***** of shatter regime multiple cratering penetration*****
0080      C          #####Shatter regime removed for independent use of ballistic#####
0081      C          #####And fracture regime evaluation#####
0082          SHATTER = .TRUE.
0083          RETURN 1
0084      END IF
0085      RETURN
0086      END

```

```
0001
0002      SUBROUTINE Fract(MR,V0,Vf,FTheta,MProj,ToverD)
0003      REAL MR,MProj
0004      IF (V0.GT.Vf+2000.) THEN
0005          MR = MProj*FTheta*.16
0006      ELSE IF (ToverD.GT..25) THEN
0007          MR = MProj*FTheta*.25
0008      ELSE
0009          MR = MProj*FTheta*.667
0010      END IF
0011      RETURN
0012      END
```

```
0001
0002      SUBROUTINE LarMR(VR,MR,V50,Thickness,
0003      :                  EffectiveThicknessRatio,Theta,Diam,ProjDensity,
0004      :                  TargetDensity,TargetYieldStrength,Penetration)
0005      REAL MR
0006      LOGICAL Penetration
0007      V50=SQRT((Thickness*EffectiveThicknessRatio/COS(Theta)/
0008      1  (0.281*Diam*(ProjDensity/TargetDensity)**.33333))
0009      2  *(1./31)/ProjDensity*(2.*TargetYieldStrength))
0010      IF (VR.LT.V50) THEN
0011          Penetration = .FALSE.
0012      ELSE
0013          Penetration = .TRUE.
0014      END IF
0015      RETURN
0016      END
```

D180-30550-4

```

0001
0002      SUBROUTINE MassErr(V1,MFC1,BHNC,RhoP,RhoT,Thick,Diam,MR)
0003      REAL MR,MS1,MFC1
0004      RhoC = RhoP*.01873
0005      RhoS = RhoT*.01873
0006      T = Thick*12.
0007      DFC1 = Diam*12
0008      C      DFC1 = SQRT(4*AC/PI)
0009      UFC=SQRT(2680*BHNC/RHOC)
0010      VP=V1/UFC
0011      AC1=PI*DFC1**2/(4*COS(THETA))
0012      MS1=RHOS*T*AC1
0013      MR=MFC1+0.5*MS1*LOG(2.74/(1+VP**2))
0014      MR = AMIN1(MR,MFC1)
0015      RETURN
0016      END

```

```

0001      C                                D180-30550-4
0002      C
0003      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0004      C
0005      C          SUBROUTINE SINGLE ( ANGR,DIA,VELE,VWTHK,DENS,FTU )
0006      C
0007      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008      C
0009      C
0010      C      Single determines the critical projectile diameter for single plate
0011      C      configurations. It is based on the Schmidt-Holsapple crater volume
0012      C      equation. The equation was solved for the critical diameter as a
0013      C      function of the plate and projectile properties in english units.
0014      C      It assumes that a hemispherical crater depth equal to 70 percent of
0015      C      the plate thickness causes failure. This is a attempt to account for
0016      C      spall and was reccomended by Mike Bjorkman of the Boeing SHock Physics
0017      C      group.
0018      C
0019      C
0020      C      Ref: 'On Scaling of the Crater Dimensions 2. Impact Processes',
0021      C      K.A. Holsapple & R.M. Schmidt, JGeophy Res, v87,nb3,10 March 1982,
0022      C      p1849-70
0023      C
0024      C
0025      C      Variable List
0026      C
0027      C      angr = impact angle measured from the normal, radians
0028      C      dia = critical projectile diameter, in
0029      C      dr = intermediate variable
0030      C      fr =      "      "
0031      C
0032      C      vele = relative velocity of the projectile, ft/sec
0033      C      vwthk = vessel wall thickness, in
0034      C
0035      C      Array list
0036      C
0037      C      dens = density, lb/in**3
0038      C      1- shield
0039      C      2- vessel wall
0040      C      3- projectile
0041      C      ftu = ultimate tensile strength, psi
0042      C      1- shield
0043      C      2- vessel wall
0044      C      3- projectile
0045      C
0046      C
0047      C      DIMENSION DENS(3),FTU(3)
0048      C
0049      C      DR=(DENS(3)/DENS(2))**(-0.159)
0050      C      V2=(VELE*COS(ANGR))**2
0051      C      FR=(2.6833*FTU(2)/DENS(3)/V2)**0.236
0052      C
0053      C      DIA=1.442*DR*FR*VWTHK
0054      C
0055      C      RETURN
0056      C
0057      C      END

```

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE PEN4 ( ANGR,BHARD,C,DENS,DIA,VWTHK,FY,FSU,FTU,
0005 1 INITIAL,SHATTER,SHPV,SHTHK,STAND,VELE )
0006 C
0007 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008 C
0009 C PEN4 DETERMINES THE DIAMETER THAT JUST PENETRATES. IT UTILIZES
0010 C THE UPDATED BOEING'S HYPERVELOCITY CODE PEN4 VERSION 10.
0011 C
0012 C
0013 C VARIABLE LIST
0014 C
0015 C ANGR = IMPACT ANGLE MEASURED FROM THE NORMAL , RADIANS
0016 C BHARD = ARRAY CONTAINING THE VALUES OF THE BRINNEL HARDNESS FOR
0017 C THE SHIELD & VESSEL WALL MATERIALS
0018 C C = ARRAY CONTAINING VALUES FOR THE SPEED OF SOUND FOR THE
0019 C SHIELD AND VESSEL WALL MATERIALS (FT/SEC)
0020 C DEN = PROJECTILE DENSITY (LB/CFT)
0021 C DENS = ARRAY CONTAINING VALUES FOR THE DENSITY FOR THE
0022 C SHIELD AND VESSEL WALL MATERIALS (LB/IN**3)
0023 C DIA = SPHERICAL PROJECTILE DIAMETER (IN)
0024 C DIA1 = PREVIOUS DIAMETER THAT PENETRATED (IN)
0025 C DIA2 = DIAMETER THAT PENETRATED TWO ITERATIONS PREVIOUSLY (IN)
0026 C EVWTHK = EQUIVALENT VESSEL WALL THICKNESS , IN.
0027 C FY = ARRAY CONTAINING THE VALUES OF THE YIELD STRENGTH FOR THE
0028 C SHIELD & VESSEL WALL MATERIALS
0029 C FSU = ARRAY CONTAINING VALUES OF THE ULTIMATE SHEAR STRENGTH
0030 C FOR THE SHIELD AND VESSEL WALL MATERIALS (PSI)
0031 C FTU = ARRAY CONTAINING VALUES OF THE ULTIMATE TENSILE STRENGTH
0032 C FOR THE SHIELD AND VESSEL WALL MATERIALS (PSI)
0033 C I = COUNTER
0034 C INITIAL = LOGICAL PARAMETER USED TO DETRMINIE IF CURRENT CALL
0035 C IS INITIAL ONE.
0036 C MASS = PROJECTILE MASS , LBS
0037 C N = INCREMENT MULIPLIER
0038 C P1 = LAST MASS GUESS TO NOT PENETRATE
0039 C P2 = LAST MASS GUESS TO PENETRATE
0040 C PEN = TRUE OR FALSE
0041 C PI = 3.14
0042 C PMINCR = INITIAL MASS GUESS INCREMENT
0043 C RATIO = P2 / P1
0044 C SHATTER = LOGICAL PARAMETER USED TO DETERMINE IF PEN4 BALLISTIC
0045 C LIMITS HAVE BEEN EXCEEDED
0046 C SHPV = ARRAY CONTAING VALUES OF THE SHOCK PROJECTILE VELOCITY
0047 C FOR THE SHIELD AND VESSEL WALL MATERIALS , ( UNITLESS )
0048 C SHTHK = SHIELD THICKNESS , IN.
0049 C SPACE = ARRAY CONTAINING THE SHIELD SPACING ,FT.
0050 C STAND = SHIELD STAND-OFF , IN.
0051 C TARMAT = ARRAY CONTAINING MATERIAL PROPERTY POINTERS
0052 C THETA = IMPACT ANGLE (DEG) , MEASURED FROM THE NORMAL
0053 C THICK = SHIELD & VESSEL WALL THICKNESS , FT.
0054 C VELE = COLLISION VELOCITY , FT/SEC
0055 C VEL1 = VEL FOR DIA1
0056 C VEL2 = VEL FOR DIA2
0057 C V0 = IMPACT VELOCITY (FT/SEC)
0058 C VR2 = RESIDUAL VELOCITY AFTER VESSEL WALL , FT/SEC
0059 C
0060 C
0061 C
0062 C DIMENSION BHARD(3),C(3),DENS(3),FY(3),FSU(3),FTU(3),PDENS(3),
0063 C PFY(3),PFSU(3),PFTU(3),SHPV(3),SPACE(2),THICK(3),

```

```

0064          TARMAT(3)
0065      C
0066          INTEGER PROJMAT,TARMAT
0067      C
0068          LOGICAL PEN,INITIAL,SHATTER
0069      C
0070          REAL MASS
0071      C
0072          SAVE DIA1,DIA2,P2,VEL1,VEL2
0073      C
0074      C
0075      C SET INITIAL VALUES
0076      C
0077          I = 0
0078          N = 0
0079          PEN = .FALSE.
0080          PMINCR = 1.0E-04
0081          PI = 3.1415926536
0082          P1 = 1.0E-06
0083      C
0084      C CONVERT VARIABLES TO PEN4 FORMAT AND UNITS
0085      C
0086      C
0087          PROJMAT=2
0088          SPACE(1) = STAND/12.0
0089          SPACE(2) = 0.0
0090          TARMAT(1) = 1
0091          TARMAT(2) = 2
0092          TARMAT(3) = 0
0093          THETA = ANGR*180.0/PI
0094          THICK(1) = SHTHK/12.0
0095          THICK(2) = VWTHK/12.0
0096          THICK(3) = 0.0
0097      C
0098      C PEN4 DOES NOT SOLVE FOR THE CRITICAL PROJECTILE DIAMETER DIRECTLY
0099      C THEREFORE AN ITERATIVE APPROACH IS TAKEN. PEN4 ONLY RETURNS
0100      C A LOGICAL PARAMETER INDICATING IF PENETRATION HAS OCCURED.
0101      C
0102      C
0103      C FOR INITIAL CASE SET P2 > 0.0
0104      C
0105          IF (INITIAL) THEN
0106              P2 = 1.0E-04
0107          ENDIF
0108      C
0109      C IF SHATTER HAS OCCURED NO NEED TO CALL PEN4 , USE LINEAR APPROX.
0110      C
0111      C
0112      C DETERMINE INITIAL MASS THAT PENETRATES
0113      C
0114      100  MASS = P2 + N * PMINCR
0115      C
0116          NC = NC + 1
0117      C
0118          CALL NPEN4 ( VELE,MASS,THETA,THICK,SPACE,PEN,SHATTER,PROJMAT,
0119      1              DENS,FTU,C )
0120      C
0121          IF ( PEN ) THEN
0122      C
0123      C IF PENETRATION TRY WITH MASS BETWEEN P1 & P2
0124      C
0125          PEN = .FALSE.
0126          P2 = MASS

```

D180-30550-4

```

0127      MASS = ( P1 + P2 ) / 2.0
0128      C
0129      NC = NC + 1
0130      CALL NPEN4 ( VELE,MASS,THETA,THICK,SPACE,PEN,SHATTER,PROJMAT,
0131      1          DENS,FTU,C )
0132      C
0133      C
0134      200      IF ( PEN ) THEN
0135      C
0136      C          IF PENETRATION , SET P2 = MASS & CHECK RATIO
0137      C
0138      C          PEN = .FALSE.
0139      C          P2 = MASS
0140      C          RATIO = P2 / P1
0141      C          IF ( RATIO .LT. .995 .OR. RATIO .GT. 1.005 ) THEN
0142      C
0143      C          IF TRUE TRY AGAIN BETWEEN P1 & P2
0144      C
0145      C          MASS = ( P1 + P2 ) / 2.0
0146      C          NC = NC + 1
0147      C          CALL NPEN4 (VELE,MASS,THETA,THICK,SPACE,PEN,SHATTER,
0148      1          PROJMAT,DENS,FTU,C )
0149      C          GO TO 200
0150      C          ELSE
0151      C
0152      C          IF FALSE , MASS IS WITHIN RANGE SO CONTINUE
0153      C
0154      C          GO TO 300
0155      C          END IF
0156      C
0157      C          ELSE
0158      C
0159      C          IF FALSE , SET P1 = MASS AND CHECK RATIO
0160      C
0161      C          P1 = MASS
0162      C          RATIO = P2 / P1
0163      C          IF ( RATIO .LT. .995 .OR. RATIO .GT. 1.005 ) THEN
0164      C
0165      C          IF TRUE , TRY AGAIN BETWEEN P1 & P2
0166      C
0167      C          MASS = ( P1 + P2 ) / 2.0
0168      C          NC = NC + 1
0169      C          CALL NPEN4 (VELE,MASS,THETA,THICK,SPACE,PEN,SHATTER,
0170      1          PROJMAT,DENS,FTU,C)
0171      C          GO TO 200
0172      C          ELSE
0173      C
0174      C          IF FALSE , MASS IS WITHIN RANGE SO CONTINUE
0175      C
0176      C          GO TO 300
0177      C          END IF
0178      C          ENDIF
0179      C          ELSE
0180      C
0181      C          IF FALSE INCREASE INCREMENT & TRY AGAIN
0182      C
0183      C          P1 = MASS
0184      C          I = I + 1
0185      C          N = 2 ** I
0186      C          GO TO 100
0187      C          END IF
0188      C
0189      C          P2=MASS

```

D180-30550-4

```
0190      C
0191      C  CALCULATE DIAMETER
0192      C
0193      C
0194      C
0195      300  DEN = DENS(3)*1728.0
0196      C
0197          DIA = ((MASS*6.0)/(PI*DEN))**(1.0/3.0)
0198      C
0199      C  CONVERT TO IN.
0200      C
0201          DIA = DIA * 12.0
0202      C
0203      C
0204      C
0205      C  RESET HOLDERS
0206      C
0207          DIA2=DIA1
0208          DIA1=DIA
0209          VEL2=VEL1
0210          VEL1=VELE
0211      C
0212      C
0213          RETURN
0214      C
0215          END
```

D180-30550-4

```
0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE NPEN4 (Vil,MProj,Thetal,Thick1,Space,Pennon,
0005 : Shater,PrMat1,Densel,YStrn1,SoundV)
0006 C
0007 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008 C
0009 C
0010 C
0011 C PrMat Integer Projectile Material 1:Steel, 2:Aluminum ,3:Ice
0012
0013 CHARACTER Shape*3
0014 LOGICAL Pennon,Shater
0015 INTEGER Maxk(5)
0016 INTEGER PrMat,PrMat1,TarMat(10),TMatSp(10),Plate,Bin,NBin,I
0017 REAL RF(5),RC(5)
0018 REAL NF,J,MR,MProj,LastSp,LRM,FrMass(5)
0019 REAL Diam,Vi,Vr,ViLRM,VrLRM,Epsil,Gamma,Vil,Pi,Theta,SumSp
0020 REAL A,B,D,R,X,Y,ToverD,Rh,PlateM,FTheta,AllMas,Vc,DelJ,DelJ2
0021 REAL P,EffP,Pet,Pgrady,Thetal,AvgMas,Rp,F1,Vf
0022 REAL Thick(10),Space(9),Thick1(10),
0023 : PDense(3),PYStrn(3),PSondV(3),FrTuff(3),
0024 : Dense(10),YStren(10),SoundV(10)
0025 REAL Densel(10),YStrn1(10)
0026 REAL ViX,MrMax,MProjX
0027 DOUBLE PRECISION Intact,HoArea
0028 DOUBLE PRECISION SumPr(5)
0029 DOUBLE PRECISION Nr,Pcr,Lambda,SigSq,Sigma,Pk,As,Ac
0030
0031 COMMON /Prob/Nr(5),Pcr(5),Lambda(5),SigSq(5),Sigma(5)
0032 COMMON /Crater/As,Ac(5),P(5)
0033 C Steel Aluminum Ice
0034 DATA PDense/ 15.11 , 5.39 , 1.94/
0035 DATA PYStrn/ 8.35E+07 , 5485000., 0.0/
0036 DATA PSondV/ 14960. , 17569., 0.0/
0037 DATA FrTuff/ 36. , 39., 39./
0038 C PrMat Projectile Material 1:Steel, 2:Aluminum ,3:Ice
0039 C Calculation units
0040 C Dense in Slugs/CubicFoot Density
0041 C YStren in Lbs/SquareFoot Uniaxial Ultimate Stress
0042 C SoundV in Feet/Second Speed of Sound
0043 C Theta in Radians Impact Angle
0044 C FrTuff in MegaPascals Meter^.5 Fracture Toughness
0045 C Input units
0046 C Densel input in Lbs/CubicInch
0047 C YStrn1 input in PSI
0048 C SoundV input in Feet/Sec
0049 C Vil Feet/Sec Impact Velocity
0050 C MProj Pounds Projectile Mass
0051 C Thetal Degrees Striking Angle
0052 C Thick Feet Target Plate Thickness
0053 C Space Feet Spacing Between Target Plates
0054 C Outputs
0055 C Pennon Logical Penetration Flag
0056 C Shater Logical Fragment Shatter Flag
0057
0058 DATA Gamma/1.5708/
0059 DATA F1/4.0/
0060 DATA NBin/5/
0061 DATA TarMat/1,2,3,4,5,6,7,8,9,10/
0062 DATA TMatSp/2,2,2,2,2,2,2,2,2,2/
0063 DO 10 Plate=1,10
```

D180-30550-4

```

0064      Thick(Plate)=Thick1(Plate)
0065      Dense(Plate)=Dense1(Plate)/32.2*1728.
0066      YStren(Plate)=YStrn1(Plate)*144.
0067      10  CONTINUE
0068      Theta = Theta1/57.3
0069      PrMat = PrMat1
0070      C    END DO
0071      Vi=Vi1
0072      PI=3.141592653589793
0073      Shater =.FALSE.
0074      C    *****Calculate Radius of Projectile Sphere*****
0075      RP=(MProj*3./(PDense(PrMat)*32.2*4.*PI))**(1./3.)
0076      Diam = RP * 2.
0077      C    *****
0078
0079      DO 2040 Plate=1,10
0080      IF (.NOT.Shater) THEN
0081      C    *** This option is for single penetrators*****
0082      C    **Compute Residual Velocity
0083
0084      Call ResVel
0085      :      (Vr,Vi,RP,PDense(PrMat),Dense(TarMat(Plate)),
0086      :      Thick(Plate),Theta,PrMat,TMatSp(Plate))
0087
0088      C    *****IF No Penetration Report Result and Exit*****
0089      IF (Vr.EQ.0.) THEN
0090      Pennon = .FALSE.
0091      RETURN
0092      ELSE IF (Thick(Plate+1).EQ.0.0.OR.Plate.EQ.10) THEN
0093      Pennon = .TRUE.
0094      RETURN
0095      ENDIF
0096      C    *****
0097      C    *****Mass Loss Regime Decision BAC IR&D*****
0098      ToverD = Thick(Plate)/Diam
0099      IF (PrMat.EQ.2.AND.TMatSp(Plate).EQ.2) THEN
0100      IF (ToverD.LT..1) THEN
0101      Vf = 1116*ToverD**(-.55)
0102      ELSE
0103      Vf = 4757*ToverD**.08
0104      END IF
0105      ELSE IF (PrMat.EQ.1.AND.TMatSp(Plate).EQ.2.) THEN
0106      IF (Shape.EQ.'CYL') THEN
0107      Vf=5020* ToverD**.4
0108      ELSE IF (Shape.EQ.'CUB') THEN
0109      Vf = AMAX1(1450*ToverD**(-.39),
0110      :      4561*ToverD**42*(Diam*12*2.54)**(-.33))
0111      ELSE
0112      Vf = AMAX1(1450*ToverD**(-.39),
0113      :      4561*ToverD**.42*(Diam*12*2.54)**(-.33) )
0114
0115      C      Vf = AMAX1(2362*ToverD**(-.35)*(Diam*12*2.54)**(-.32),
0116      C      :      3937*ToverD**.23 *(Diam*12*2.54)**(-.25))
0117      END IF
0118      ELSE IF (PrMat.EQ.1.AND.TMatSp(Plate).EQ.1.) THEN
0119      Vf = 7021*ToverD**.39
0120      END IF
0121      C    *****
0122      C    RH=RP*(1.372E-4*Vi*(THICK(Plate)/(2.*RP))**(2./3.)+.9)*
0123      C    :      (1-EXP(-(1.48-Theta)/.166))
0124      RH = .5*Thick(Plate)*11.02*(1-EXP(-(1.48-Theta)/.166))*
0125      :      (1-EXP(-(PDense(PrMat)*Vi**2/YStren(TarMat(Plate))))**.415
0126      :      *(PDense(PrMat)/Dense(TarMat(Plate))))**(-.15)/ToverD/29.9)

```

```

0127      RH = AMAX1(RH,RP)
0128      IF (Vi.GT.Vf) THEN
0129          SumSp=Space(Plate)
0130          Shater=.TRUE.
0131      C      *** Compute plate spall *****
0132          PlateM=PI*RH**2/COS(Theta)*Thick(Plate)*
0133              :      Dense(TarMat(Plate))*32.2
0134      C      *****
0135      C      *****Hydrocode Predicted Mass Loss Due to Impact Angle*****
0136          FTHETA=-2.433E-4 - 1.643E-2*COS(THETA)+3.201*COS(THETA)**2-
0137              :      2.184*COS(THETA)**3
0138      C      *****
0139      C      *** COMPUTE FRAGMENT NUMBERS*****
0140          IF (PrMat.EQ.3) THEN
0141              MProj=PlateM
0142              RP=(PlateM*3./ (Dense(TarMat(Plate))*32.2*4.*PI))**(1./3.)
0143          END IF
0144
0145          CALL MasChr (Vi,MProj,ToverD,Theta,RP,PDense(PrMat),
0146              :      Dense(TarMat(Plate)),PSondV(PrMat),NBin,
0147              :      RF,NR,LRM,AvgMas,PrMat,FrTuff(PrMat))
0148
0149          IF (PrMat.EQ.3) THEN
0150              PrMat=TMatSp(Plate)
0151          ELSE
0152              AllMas=PlateM+MProj
0153              DO 2010 I=1,NBin-1
0154                  FrMass(I)=4./3.*PI*RF(I)**3*PDense(PrMat)*32.2
0155                  NR(I)=NR(I)*AllMas/MProj
0156      2010      CONTINUE
0157          END IF
0158          Nr(NBin) = 1.D0
0159          ViLRM=Vr
0160      C      *****Calculate Spray Angle*****
0161      C      Vc =11155*ToverD**(-.52)
0162      C      Epsil = 45*(1-EXP((- (Vi-Vf)/Vc)))/57.3
0163      C      Vc = 4889 * ToverD**(-.23)
0164      C      Epsil = 52*(1-EXP((- (Vi-Vf)/Vc)))/57.3
0165      C      *** Assurance of ThetaR+Epsil<90 and Spray Area finite*****
0166      C      ThetaR = AMIN1(Theta,1.41-Epsil)
0167      C      *** Calculate Spray Area*****
0168          DELJ=RP/2.0*(COS(EPSIL)-TAN(ThetaR)*SIN(EPSIL))
0169          DELJ2=RP*(1.0-TAN(ThetaR)*TAN(EPSIL))
0170          J=Space(Plate)*SIN(GAMMA)/SIN(ThetaR+GAMMA)
0171          X=(J*SIN(EPSIL)/2.+DELJ)*(1./SIN(ThetaR+GAMMA+EPSIL)-
0172              :      1./SIN(ThetaR+GAMMA-EPSIL))
0173          Y=X*SIN(ThetaR+GAMMA)
0174          R=(J-X*COS(ThetaR+GAMMA))*TAN(EPSIL)+DELJ2
0175          B=SQRT(R**2-Y**2)
0176          A=(J*SIN(EPSIL)/2.+DELJ)*(1./SIN(ThetaR+GAMMA-
0177              :      EPSIL)+1./SIN(ThetaR+GAMMA+EPSIL))
0178          AS=PI*A*B
0179      C      *****
0180      C      *** Allowance for increased penetration due to spalling of ***
0181      C      *** next plate *****
0182          EffP=1.7
0183      C      *****
0184      ELSE
0185          IF (PrMat.EQ.3) THEN
0186              MProj=3.1415926*Rh**2*Thick(Plate)
0187              :      *Dense(TarMat(Plate))*32.2
0188              PrMat=TMatSp(Plate)
0189          END IF

```

```

0190      Vi=Vr
0191      END IF
0192      C      *****
0193      ELSE
0194
0195      C      *****The rest of this subroutine contains the evaluation*
0196      C      ***** of shatter and fracture regime multiple cratering*
0197
0198      C      *** Allowance for increased penetration due to spalling of ****
0199      C      *** next plate *****
0200      Thick(Plate) = Thick(Plate)/EffP
0201      C      *****
0202
0203      C      *** COMPUTE PLATE CRATER Depth and Radius*****
0204      DO 2030 I=1,NBin
0205          D=2.*RF(I)
0206          PGrady=0.281*D*(PDense(PrMat)/Dense(TarMat(Plate)))*(1./3.)
0207          :      *(PDense(PrMat)*(Vr*COS(Theta))**2/
0208          :      (2.*YStren(TarMat(Plate)))*0.31
0209          IF (PrMat.EQ.1) THEN
0210              Pet = D*(Vr*COS(Theta)/12468.)*1.32
0211              P(I) = AMIN1(Pet,PGrady)
0212              IF (Vr.LT.3.9*3281) THEN
0213                  RC(I) = P(I)/2.*(Vr/3281./3.8)**(-1.32)
0214              ELSE
0215                  RC(I) = P(I)/2.*(Vr/3281./4.6)**.2
0216              END IF
0217          ELSE
0218              Pet = D*2.33E-5*(Vr*COS(Theta))*1.16
0219              P(I) = AMIN1(Pet,PGrady)
0220              RC(I) = P(I)/(1-EXP(-Vr/5578.))
0221          END IF
0222      C      *****
0223
0224      C      *** COMPUTE AVERAGE IMPACTS WITHIN CRATER*****
0225      AC(I)=PI*RC(I)**2/F1
0226      AC(I)=DMIN1(AC(I),.9999999999999999D0*AS)
0227      PCR(I)=DMIN1(1D0,AC(I)/AS)
0228      LAMBDA(I)=NR(I)*AC(I)/AS
0229      SigSq(I) = Lambda(I)*(1D0-PCR(I))
0230      Sigma(I) = SQRT(SigSq(I))
0231      C      *****
0232      2030      CONTINUE
0233      C      *** New Version eight section*****
0234      C      *** This subroutine finds the minimum number of each***
0235      C      *** size particle that must impact in one crater to ***
0236      C      *** penetrate the plate ,how many craters they are in,*
0237      C      *** and how many fragments are involved in shallower***
0238      C      *** craters.*****
0239      CALL PenK(Plate,Thick,NBin,Maxk)
0240      CALL Countr
0241      :      (NBin,P,Thick(Plate),Maxk,Intact,Nr,Ac,As)
0242      C      *** Number and Area of Holes and Residual Particles****
0243      HoArea=As*(1.D0-Intact)
0244      PlateM = HoArea*Thick(Plate)*Dense(Plate)*32.2
0245      AllMas=0.
0246      DO 2110 I=1,NBin-1
0247          AllMas=FrMass(I)*Nr(I)+AllMas
0248      2110      CONTINUE
0249      IF (AllMas.GT.0) THEN
0250          DO 2210 I=1,NBin-1
0251              NR(I)=NR(I)*(1.+PlateM/AllMas)
0252      2210      CONTINUE

```

D180-30550-4

```

0253      END IF
0254      C      *** Separate calculation for LRM***
0255      C      *** This option is for single penetrators*****
0256
0257      Call ResVel
0258      :      (Vr,ViLRM,RF(NBin),PDense(PrMat),Dense(TarMat(Plate)),
0259      :      Thick(Plate),Theta,PrMat,TMatSp(Plate))
0260
0261      C      ** Convert V to Km/S and MProj to grams
0262      ViX = Vi/3281.
0263      MProjX = LRM * 454.
0264
0265      C      ** Largest Residual Mass **
0266      ToverD=Thick(Plate)/2./Rf(NBin)
0267      CALL RFMax(MProjX,PDense(PrMat),ViX,RF(NBin),ToverD,MRMax,
0268      :      PSondV(PrMat),PrMat,Vc,Theta,FrTuff(PrMat))
0269      LRM=MrMax/454
0270      RF(NBin)=(LRM*3./(PDense(PrMat)*32.2*4.*PI))**(1./3.)
0271
0272      C      *****
0273      C      *** Test for Pennon and End of Run*****
0274      IF (HoArea.LT..000000069.AND.Vr.LE.0.00001) THEN
0275      C      IF (HoArea.LT.AC(1)) THEN
0276      Pennon = .FALSE.
0277      RETURN
0278      ELSE IF (Plate.EQ.10.OR.Thick(Plate+1).EQ.0.0) THEN
0279      Pennon = .TRUE.
0280      RETURN
0281      END IF
0282      ViLRM=Vr
0283      LastSp=SumSp
0284      SumSp=SumSp+Space(Plate)
0285      As=As*(SumSp/LastSp)**2
0286      C      *****
0287      C      *****END OF SHATTER EVALUATION*****
0288      END IF
0289      2040 CONTINUE
0290      RETURN
0291      END

```

D180-30550-4

```

0001
0002 SUBROUTINE RFMax(M,RhoP,V,RP,ToverD,MrMax,C,PrMat,Vc,Theta,Kic)
0003 C M Initial Impactor Mass Grams
0004 C MrMax Mass of the Largest Residual Particle Same as above
0005 C RhoP Impactor Density Slugs/Ft^3
0006 C Rp Initial Impactor Radius (equivalent sphere) Ft
0007 C V Impactor Velocity Km/Sec
0008 C Shock Velocity Ft/Sec
0009 C Kic Fracture Toughness MPa m^.5
0010 C Theta Approach angle Radians
0011 C PrMat Projectile Material 1:Steel, 2:Aluminum ,3:Ice
0012 REAL Kic,k,MrMax,M,MrOMs,MrOMsC
0013 REAL MrMaxP,MrOMsP
0014 REAL MrMaxT,MrOMsT
0015 REAL RhoP,V,RP,ToverD,C,FTovrD,Vc
0016 INTEGER PrMat
0017 k=4.18E6
0018 IF (ToverD.LT..1) THEN
0019 FTovrD=1180.
0020 ELSE IF (ToverD.GE..1.AND.ToverD.LT..2) THEN
0021 FTovrD=697*ToverD**(-.23)
0022 ELSE IF (ToverD.GE..2.AND.ToverD.LT..4) THEN
0023 FTovrD=244*ToverD**(-.881)
0024 ELSE IF (ToverD.GE..4.AND.ToverD.LT..8) THEN
0025 FTovrD=1500*ToverD**1.1
0026 ELSE
0027 FTovrD=1170.
0028 END IF
0029 IF (PrMat.EQ.1) THEN
0030 FTovrD=FTovrD*7.78
0031 END IF
0032 MrOMs = k*(Kic/(RhoP*32.2*C))**2/V**2/(RP*2*12*2.54)*FTovrD
0033 MrOMsT = MrOMs*COS(Theta)**(-2)
0034 MrOMsP = MrOMs*COS(Theta)
0035 Vc =(8.47E4*MrOMs*V**2)**(1/10.93)
0036 IF (V.LT.Vc) THEN
0037 MRMaxP=M*MrOMsP
0038 ELSE
0039 MrOMsC=1.18E-5*(Vc)**8.93
0040 MrMaxP=M*MrOMsC*(V/Vc)**(-5.5)
0041 END IF
0042 IF (V*COS(Theta).LT.Vc) THEN
0043 MRMaxT=M*MrOMsT
0044 ELSE
0045 MrOMsC=1.18E-5*(Vc)**8.93
0046 MrMaxT=M*MrOMsC*(V*COS(Theta)/Vc)**(-5.5)
0047 END IF
0048 MrMax = AMAX1(MrMaxP,MrMaxT)
0049 IF (PrMat.EQ.1) MrMax=MrMaxP
0050 MrMax = AMIN1(MrMax,M*.9999)
0051 RETURN
0052 END

```

D180-30550-4

```

0001
0002 SUBROUTINE PenK(Plate,Thick,NBin,Maxk)
0003 DOUBLE PRECISION Nr,Pcr,Lambda,SigSq,Sigma,Pk(5),SumPr(5),Pk2
0004 DOUBLE PRECISION As,Ac,TestAc
0005 INTEGER Bin,Plate,MaxK(5),I,BinsDo,NBin
0006 REAL Thick(10),P,Vc
0007 LOGICAL BINDON(5)
0008 INTEGER k(5),MinI
0009 DOUBLE PRECISION kProbs(5,0:15)
0010 REAL LnPk,LnkFac
0011 COMMON /Count/kProbs
0012 COMMON /Prob/Nr(5),Pcr(5),Lambda(5),SigSq(5),Sigma(5)
0013 COMMON /Crater/As,Ac(5),P(5)
0014
0015 C *** k(NBin) is the number of each particle size that must ****
0016 C *** impact in one crater to make a hole.*****
0017 C *** This loop finds the fraction of the plate that is not hit by *****
0018 C *** a fragment of size I (P(0)). BinDon are initialized***
0019 TestAc=1D0-1D-3*Ac(1)/As
0020 BinsDo=0
0021 DO 10 I=1,NBin
0022     k(I)=0D0
0023     CALL Prs(I,Pk(I),k,NBin)
0024     kProbs(I,0)=Pk(I)
0025     SumPr(I)=Pk(I)
0026 C *** If there is not room for one more crater on the plate then *****
0027 C *** stop using this particle size *****
0028     IF (SumPr(I).GT.TestAc.OR.
0029 :         (K(I)*P(I)).GT.Thick(Plate)) THEN
0030         BinDon(I)=.TRUE.
0031         BinsDo=BinsDo+1
0032         kProbs(I,1)=1D0-kProbs(I,0)
0033     ELSE
0034         BinDon(I)=.FALSE.
0035     END IF
0036 C *****
0037 10 CONTINUE
0038 C END DO
0039 C *****
0040 C *** This subroutine finds which fragment size has the least SumPr *****
0041 C *** SumPr is the fraction of the plate that has craters of depth <= k**
0042 C *****
0043 C *** This loop sums up the area of the plate that is not penetrated,****
0044 C *** while keeping the area covered by craters of depth <=k for each ***
0045 C *** size approximately equal.*****
0046 20 IF (BinsDo.GE.NBin-1) GOTO 30
0047     CALL MiniI(MinI,BinDon,NBin,SumPr)
0048     k(MinI) = k(MinI)+1
0049 C *** This subroutine Calculates the fraction of the plate that is ****
0050 C *** covered by craters from exactly k particles of size MinI *****
0051     CALL PrS(MinI,Pk(MinI),k,NBin)
0052 C *****
0053 C *** Add up the fraction of the spray area accounted for, and the ****
0054 C *** fraction of the particles used so far*****
0055     SumPr(MinI)=SumPr(MinI)+Pk(MinI)
0056     kProbs(MinI,k(MinI))=Pk(MinI)
0057 C *****
0058 C *** If there is not room for one more crater on the plate then *****
0059 C *** stop using this particle size*****
0060     IF (SumPr(MinI).GT.TestAc.OR.
0061 :         (K(MinI)*P(MinI)).GT.Thick(Plate)) THEN
0062         BinDon(MinI)=.TRUE.
0063         BinsDo=BinsDo+1

```

D180-30550-4

```

0064      Maxk (MinI) =K (MinI)
0065      END IF
0066 C      *****
0067      GOTO 20
0068 30     CONTINUE
0069      K(NBin) = 1
0070      kProbs (NBin,1)=1D0-kProbs (NBin,0)
0071      Maxk (NBin) = 1
0072 C      *****
0073      RETURN
0074      END

```

D180-30550-4

```

0001
0002      SUBROUTINE MiniI(MinI,BinDon,NBin,SumPr)
0003      LOGICAL BinDon(5)
0004      INTEGER BinsDo
0005      DOUBLE PRECISION SumPr,MinSum
0006      DIMENSION SumPr(5)
0007      MinSum=1D0
0008      DO 10 I=1,NBin-1
0009          IF (.NOT.BinDon(I).AND.SumPr(I).LT.MinSum) THEN
0010              MinI=I
0011              MinSum=SumPr(I)
0012          END IF
0013      10 CONTINUE
0014      C      END DO
0015      RETURN
0016      END

```

```

0001
0002      SUBROUTINE PrS(I,Pk,k,NBin)
0003      DOUBLE PRECISION LastTe(5),PkMin,Fpi
0004      DOUBLE PRECISION Nr,Pcr,Lambda,SigSq,Sigma,Pk,SumPr
0005      DOUBLE PRECISION As,Ac
0006      DIMENSION k(5)
0007      DIMENSION PkMin(5),kMin(5)
0008      COMMON /Prob/Nr(5),Pcr(5),Lambda(5),SigSq(5),Sigma(5)
0009      Fpi=1D0/SQRT(2D0*3.1415926D0)
0010      IF (Nr(I).GE.50.AND.Lambda(I).LT.5) THEN
0011      C      *** This section calculates the poisson approximation to ***
0012      C      *** the binomial distribution*****
0013      IF (K(I).EQ.0) THEN
0014      C      *** Initialize P(k) for first value*****
0015      Pk=DEXP(-Lambda(I))
0016      C      *****
0017      ELSE
0018      C      *** Calculate P(k) from P(k-1)*****
0019      Pk=LastTe(I)*Lambda(I)/K(I)
0020      C      *****
0021      END IF
0022      LastTe(I)=Pk
0023      C      *****
0024      ELSE IF (Pcr(I).GT..1.AND.Pcr(I).LT..9.AND.
0025      :      Nr(I).GT.9./(Pcr(I)*(1-Pcr(I)))) THEN
0026      C      *** This section calculates the normal approximation to ****
0027      C      *** the binomial distribution*****
0028      Pk=Fpi/Sigma(I)/DEXP(1/(2.D0*SigSq(I)*(k(I)-Lambda(I))**2))
0029      C      *****
0030      ELSE
0031      C      ***This section calculates the binomial distribution*****
0032      IF (k(I).EQ.0) THEN
0033      C      *** Find least non-zero P(k) and corresponding k*****
0034      CALL SetBin(I,PkMin,kMin,NBin)
0035      LastTe(I)=PkMin(I)
0036      IF (kMin(I).EQ.0) Pk=PkMin(I)
0037      C      *****
0038      ELSE IF (k(I).GT.kMin(I)) THEN
0039      C      *** calculate the non-zero P(k)s*****
0040      Pk = LastTe(I)*Pcr(I)/(1D0-Pcr(I))*(Nr(I)-k(I)+1D0)/k(I)
0041      LastTe(I)=Pk
0042      C      *****
0043      ELSE IF (k(I).EQ.KMin(I)) THEN
0044      Pk = PkMin(I)
0045      ELSE
0046      Pk = 0D0
0047      END IF
0048      C      *****
0049      END IF
0050      RETURN
0051      END

```

D180-30550-4

```

0001
0002      SUBROUTINE SetBin(I,PkMin,kMin,NBin)
0003      C      *** The Probability of the mean case occuring is computed *****
0004      C      *** in subroutine Binomial.  P(kMin) is calculated from *****
0005      C      *** this P(Lambda).  The magnitude of P(kMin) is arbitrarily ***
0006      C      *** chosen to be about 1*10^-7.*****
0007
0008      COMMON /Prob/Nr(5),Pcr(5),Lambda(5),SigSq(5),Sigma(5)
0009
0010      DOUBLE PRECISION Nr,Pcr,Lambda,SigSq,Sigma,Pk,SumPr,N1,PkMin,P1
0011      DIMENSION PkMin(5),kMin(5)
0012      k=Lambda(I)
0013      N1=Nr(I)
0014      P1=Pcr(I)
0015      CALL Binomi(k,N1,Pk,P1)
0016      1  IF (.NOT.(Pk.GT.1D-13.AND.k.GE.1)) GOTO 10
0017          Pk = Pk*(1D0-Pcr(I))/Pcr(I)*k/(Nr(I)-k-1D0)
0018          k = k-1
0019      GOTO 1
0020      10  CONTINUE
0021      C      END DO
0022          kMin(I)=k
0023          PkMin(I)=Pk
0024      RETURN
0025      END

```

D180-30550-4

```

0001
0002      SUBROUTINE Binomi (k1,Nr,Pk,Pcr)
0003      DOUBLE PRECISION Nr
0004      DOUBLE PRECISION Pk,Pcr,Lambda,LastPk,Qcr
0005      DOUBLE PRECISION Top,Lower1,Lower2,k,Expon
0006      Lambda=k1
0007      k=k1
0008      C      *****Binomial Distribution, Calculates P(k)*****
0009      Lower1 = 0d0
0010      Lower2 = 0d0
0011      Top = Nr-k
0012      Pk = 1d0
0013      Qcr = 1D0-Pcr
0014      LastPk = 0D0
0015      IF (Qcr.EQ.1.D0) THEN
0016          IF (k.EQ.0.D0) THEN
0017              Pk=1.D0
0018          ELSE
0019              Pk=0.D0
0020          ENDIF
0021      ELSE
0022          3040      IF (Lower1+Lower2+Top.GE.(2*Nr).OR.Pk.EQ.LastPk) GOTO 3030
0023                  LastPk = Pk
0024          3060      IF (Top.GE.Nr.OR.Pk.GE.1E23) GOTO 3050
0025                  Top = Top + 1D0
0026                  Pk = Pk * Top
0027                  GOTO 3060
0028          3050      CONTINUE
0029          3080      IF (Lower1.GE.k.OR.Pk.LE.1E-20) GOTO 3070
0030                  Lower1 = Lower1 + 1d0
0031                  Pk = Pk /Lower1*Pcr
0032                  GOTO 3080
0033          3070      CONTINUE
0034                  IF (Lower1.GE.k.AND.Lower2.LT.Nr-k.AND.Pk.GT.1D-27) THEN
0035                      Expon=DMIN1((-28.D0-DLOG10(Pk))/DLOG10(Qcr)),(Nr-k-Lower2))
0036                      IF (Expon.GT.0) Pk=Pk*Qcr**Expon
0037                      Lower2 = Lower2+Expon
0038                  ENDIF
0039                  GOTO 3040
0040          3030      CONTINUE
0041      END IF
0042      C      *****
0043      RETURN
0044      END

```

C - 3

D180-30550-4

```

0001
0002      SUBROUTINE MasChr (V,M,ToverD,Theta,RP,RhoP,RhoT,C,NBin,
0003      :                   MenRad,Nr,MrMax,MAvg,PrMat,FrTuff)
0004      DOUBLE PRECISION NR(5)
0005      REAL MProj,M,MRMax,MAvg,MenRad(5),MPlate
0006      REAL MasLim,NrmF,NrmS,BinMen(5),BinMas(0:5)
0007      INTEGER Bin,PrMat
0008      PI = 3.14159
0009
0010      C      This subroutine divides the residual mass into bins of equal mass
0011      C      The number of fragments in each bin are also noted
0012
0013      C      ** Convert V to Km/S and MProj to grams
0014      Vi = V/3281.
0015      MProj = M * 454.
0016      MPlate = M * 454.
0017
0018      C      ** Largest Residual Mass **
0019
0020      IF (PrMat.EQ.3) THEN
0021          CT = 17569
0022          RhoT = 5.39
0023          CALL RFMax(MPlate,RhoT,Vi,RP,ToverD,MRMax,
0024      :              CT,PrMat,Vc,Theta,FrTuff)
0025      ELSE
0026          CALL RFMax(MProj,RhoP,Vi,RP,ToverD,MRMax,
0027      :              C,PrMat,Vc,Theta,FrTuff)
0028      END IF
0029
0030      C      ** Average Residual Mass **
0031      CALL AvgRes (ToverD,Theta,MPlate,Vi,Alfa,MAvg)
0032      IF (Vi.GT.Vc) MAvg=MAvg*(Vi/Vc)**(-5.5)
0033
0034      C      ** Parameters and Normalization Constants for Weibull Distribution **
0035      CALL ShCons (ToverD,Theta,MProj,Vi,MAvg,
0036      :              bS,sS,bF,sF,NrmS,NrmF,MRMax)
0037
0038      C      ** Size Shatter Begins **
0039      CALL Shhold (MProj,Vi,TD,Theta,MAvg,bF,sF,bS,sS,
0040      :              NrmF,NrmS,FrgLim,MasLim,MRMax)
0041
0042      C      ** Upper Bound and Mean Fragment weight for each Bin **
0043      CALL BinLim(NBin,MAvg,MRMax,BinMas,BinMen,MProj,
0044      :              sF,bF,sS,bS,NrmS,NrmF,MasLim)
0045
0046      C      ** Loop Determining Number of Fragments in each Bin in Shatter Regime *
0047      DO 10 Bin = 1,NBin-1
0048          Nr(Bin)=(BinMas(Bin)-BinMas(Bin-1))/BinMen(Bin)
0049      10 CONTINUE
0050      C      END DO
0051      Nr(NBin) = 1
0052      C      ** Convert Bin Masses into lbs from grams then to Radius in Feet**
0053      DO 20 Bin = 1,NBin
0054          BinMen(Bin)=BinMen(Bin)/454.
0055          MenRad(Bin) = (BinMen(Bin)*3./ (RhoP*32.2*4.*PI))**(1./3.)
0056      20 CONTINUE
0057      C      END DO
0058      MrMax=MrMax/454.
0059      MPlate = MPlate/454.
0060      MProj = MProj/454.
0061      RETURN
0062      END

```

D180-30550-4

```

0001
0002      SUBROUTINE MasDis (M,MAvg,b,s,NrmC,MasSum)
0003      C
0004      C      THIS PROGRAM COMPUTES SHATTER Mass distributions
0005      C      STEEL CUBES ON ALUMINUM PLATE DATA.
0006      C      V = IMPACT VELOCITY      (FEET/SECOND)
0007      C      TD = T OVER D
0008      C      M = PROJECTILE MASS      (GRAINS)
0009      C      THETA = ANGLE OF OBLIQUITY
0010      C      ALPHA=ANGLE OF IMPACT
0011      C
0012      REAL MAvg,M,NrmC,MasSum
0013      W = M/MAvg
0014      C      **Mass in Bin**
0015      determ=b+s*LOG(W)
0016      IF (Determ.LT.-80) THEN
0017          MasSum=0
0018      ELSE IF (Determ.LT.4) THEN
0019          MasSum = NrmC*(1-EXP(-EXP(Determ)))
0020      ELSE
0021          MasSum=NrmC
0022      END IF
0023      RETURN
0024      END

```

D180-30550-4

```

0001      C      SSSSSSSSSSSSSS  SALVO's      CHANGES  SSSSSSSSSSSSSS
0002
0003      SUBROUTINE ShCons(TD,Theta,M,Vi,MAvg,
0004      :                      bS,sS,bF,sF,NrmCS,NrmCF,LRM)
0005      REAL M,NrmCS,NrmCF,MT2,MAvg,IntegS,IntegF,LRM
0006      sS=1.8-.04*TD/COS(THETA)-.042*M
0007      &      +.34*(COS(2.*ALPHA)**2.)-(1-EXP(-.64*Vi))
0008      bS=-2.3-1.1*TD/COS(THETA)+.0675*M
0009      &      -.27*Vi+1.4*(COS(2.*ALPHA)**2.)
0010      sF=1.38-.510*TD/COS(THETA)+.036*M
0011      &      +3.31*(COS(2.*ALPHA)**2.)-(1-EXP(-.390*Vi))
0012      bF=-1.17+.313*TD/COS(THETA)+.0675*M
0013      &      +.508*Vi-1.41*(COS(2.*ALPHA)**2.)
0014
0015      C      ** Total mass in shatter regime **
0016      MT2 =(-.957+EXP(-.0013*M))*(Vi**(.38*M+2.5))
0017      MT2 = AMAX1(MT2,1E-20)
0018      MT2 = AMIN1(MT2,M)
0019      C      **Normalization to Largest Residual Mass**
0020      C      IntegS = 1-EXP(-EXP(bS)*(M/MAvg)**sS)
0021      C      IntegF = 1-EXP(-EXP(bF)*(M/MAvg)**sF)
0022      C      NrmCS = MT2/IntegS
0023      C      NrmCF = M /IntegF
0024      detrm1=bS+sS*ALOG(LRM/MAvg)
0025      IF (Detrm1.LT.-15) THEN
0026          NrmCS=1e-30
0027      ELSE IF (Detrm1.LT.4) THEN
0028          IntegS = 1-EXP(-EXP(Detrm1))
0029          NrmCS = MT2/IntegS
0030      ELSE
0031          NrmCS = MT2
0032      END IF
0033      detrm2=bF+sF*LOG(LRM/MAvg)
0034      IF (Detrm2.LT.-15) THEN
0035          NrmCS=1e-30
0036      ELSE IF (Detrm2.LT.4) THEN
0037          IntegF = 1-EXP(-EXP(Detrm2))
0038          NrmCF = M/IntegF
0039      ELSE
0040          NrmCF = M
0041      END IF
0042      if(sF)1,2,2
0043      1      sF=0.
0044      2      if(sS)3,4,4
0045      3      sS=0.
0046      4      RETURN
0047      END

```

D180-30550-4

```

0001
0002      SUBROUTINE Shhold (MProj,Vi,TD,Theta,MAvg,bF,sF,bS,sS,
0003      :                   NrmF,NrmS,FrgLim,MasLim,LRM)
0004      REAL LowLim,Mean,Mfs,Mss,MAvg,MasLim,MProj,NrmF,NrmS,M,LRM
0005      HiLim = MProj
0006      LowLim = 0
0007      DO 10 I=1,20
0008          Mean = (HiLim+LowLim)/2
0009          CALL MasDis (Mean,MAvg,bF,sF,NrmF,Mfs)
0010          CALL MasDis (Mean,MAvg,bS,sS,NrmS,Mss)
0011          IF (Mfs.GT.Mss) THEN
0012              HiLim = Mean
0013          ELSE
0014              LowLim = Mean
0015          END IF
0016      10  CONTINUE
0017      C    END DO
0018          FrgLim = Mean
0019          MasLim = AMIN1 (Mss,MProj-LRM)
0020          RETURN
0021      END

```

D180-30550-4

```

0001
0002 SUBROUTINE BinLim (NBin,MAvg,LRM,BinMas,BinMen,MProj,
0003 : sF,bF,sS,bS,NrmS,NrmF,MasLim)
0004 REAL MAVg,LRM,MProj,MasLim,NrmS,NrmF,BinInt,MasMen,MFrac(0:4)
0005 DIMENSION BinMen(5),BinMas(0:5),MasMen(5)
0006 INTEGER Bin,ShtBin
0007 DATA MFrac/0.,.25,.5,.75,1./
0008 ShtBin=0
0009 DO 5 Bin=1,NBin-1
0010     BinMas(Bin) = (MProj-LRM)*MFrac(Bin)
0011     MasMen(Bin) = (MProj-LRM)*(MFrac(Bin)+MFrac(Bin-1))/2.
0012     IF (BinMas(Bin).LE.MasLim) ShtBin=Bin
0013 5 CONTINUE
0014 C END DO
0015 C ** Bin Limit and Mean for Shatter **
0016 DO 10 Bin = 1,ShtBin
0017     BinMen(Bin) = MAVg*(-EXP(-bS)*DLOG(1D0-MasMen(Bin)/NrmS))**(1/sS)
0018 10 CONTINUE
0019 C END DO
0020 C ** Bin Limit and Mean for First Bin in Fracture **
0021 Bin = ShtBin+1
0022 IF (Bin.LE.NBin-1) THEN
0023     IF (MasLim.GE.MasMen(Bin)) THEN
0024 C         ** Bin Mean in Case it is Less Than the Fracture Threshold **
0025         BinMen(Bin)=MAvg*(-EXP(-bS)*DLOG(1D0-MasMen(Bin)/NrmS))**(1/sS)
0026     ELSE
0027         BinMen(Bin)=MAvg*(-EXP(-bF)*DLOG(1D0-MasMen(Bin)/NrmF))**(1/sF)
0028     END IF
0029 END IF
0030 C ** Bin Limit and Mean for Fracture **
0031 DO 20 Bin = ShtBin+2,NBin-1
0032     BinMen(Bin) = MAVg*(-EXP(-bF)*DLOG(1D0-MasMen(Bin)/NrmF))**(1/sF)
0033 20 CONTINUE
0034 C END DO
0035 C ** Bin Mean for largest bin is LRM (Largest Residual Mass) **
0036 BinMen(NBin) = LRM
0037 RETURN
0038 END

```

D180-30550-4

```

0001
0002      SUBROUTINE AvgRes (ToverD,Theta,M,V,Alfa,MAvg)
0003      REAL MAVg,M
0004      C      ** Average Fragment Mass **
0005      MAVg = .0109-.00879*ToverD/COS(Theta)+.000506*M-.00428*V+
0006      :      .0110*COS(2*Alfa)**2
0007      MAVg = AMax1(MAvg,0.005)
0008      RETURN
0009      END

```

```

0001
0002      SUBROUTINE Countr(NBin,P,Thick,Maxk,Intact,Nr,Ac,As)
0003      INTEGER Bin,NBin,SmBin
0004      INTEGER Digit(5),HoMin(5),I,Maxk(5)
0005      REAL Thick,HoDept,P(5)
0006      DOUBLE PRECISION Probs(5,0:15)
0007      DOUBLE PRECISION Intact,ITArea,ITArS1
0008      DOUBLE PRECISION Nr(5),Ac(5),As
0009      DOUBLE PRECISION ArHol1,ArHol2,ArHol3
0010      DOUBLE PRECISION ArHol4,ArHol5
0011      DOUBLE PRECISION PrTemp,DigTp1,DigTp2,DigTp3
0012      DOUBLE PRECISION          DigTp4,DigTp5
0013      LOGICAL UnInc,OBinld
0014      COMMON /Count/Probs
0015      DO 10 Bin = 1,NBin
0016          HoMin(Bin) = JMIN0(INT(Thick/P(Bin)+1),Maxk(Bin))
0017          Digit(Bin)=0
0018      10  CONTINUE
0019      C      END DO
0020          ArHol1=0.D0
0021          ArHol2=0.D0
0022          ArHol3=0.D0
0023          ArHol4=0.D0
0024          ArHol5=0.D0
0025          Intact=0.D0
0026          Bin = 0
0027          HoDept = 0
0028          OBinld = .FALSE.
0029      C      DO WHILE (Bin.LE.NBin)
0030          2      IF (.NOT.(Bin.LE.NBin)) GOTO 20
0031              IF (HoDept.GT.Thick) THEN
0032                  OBinld = .FALSE.
0033                  SmBin=1
0034          C      DO WHILE (Digit(SmBin).EQ.0)
0035              21      IF (.NOT.(Digit(SmBin).EQ.0)) GOTO 210
0036                  SmBin=SmBin+1
0037                  GOTO 21
0038          210      CONTINUE
0039          C      END DO
0040              Digit(SmBin)=0
0041              HoDept=0.
0042              Bin = SmBin+1
0043              UnInc = .TRUE.
0044          C      DO WHILE (Bin.LE.NBin.AND.UnInc)
0045              22      IF (.NOT.(Bin.LE.NBin.AND.UnInc)) GOTO 220
0046                  IF (Digit(Bin).LT.HoMin(Bin)) THEN
0047                      Digit(Bin)=Digit(Bin)+1
0048                      UnInc = .FALSE.
0049                  ELSE
0050                      Digit(Bin)=0
0051                      Bin = Bin + 1
0052                  END IF
0053                  GOTO 22
0054          220      CONTINUE
0055          C      END DO
0056              DO 230 I=SmBin+1,NBin
0057                  HoDept=HoDept+Digit(I)*P(I)
0058          230      CONTINUE
0059          C      END DO
0060      ELSE
0061          IF (OBinld) THEN
0062              DigTp1=DigTp1+1
0063              IArea = Probs(1,DigTp1)*ITArS1

```

D180-30550-4

```

0064
0065      ArHol1 = ArHol1+ITArea*DigTp1
0066      ArHol2 = ArHol2+ITArea*DigTp2
0067      ArHol3 = ArHol3+ITArea*DigTp3
0068      ArHol4 = ArHol4+ITArea*DigTp4
0069      ArHol5 = ArHol5+ITArea*DigTp5
0070
0071      ELSE
0072          IArea=1.D0
0073
0074          DigTp2=Digit(2)
0075          PrTemp = Probs(2,DigTp2)
0076          IArea = PrTemp*ITArea
0077
0078          DigTp3=Digit(3)
0079          PrTemp = Probs(3,DigTp3)
0080          IArea = PrTemp *ITArea
0081
0082          DigTp4=Digit(4)
0083          PrTemp = Probs(4,DigTp4)
0084          IArea = PrTemp *ITArea
0085
0086          DigTp5=Digit(5)
0087          PrTemp = Probs(5,DigTp5)
0088          IArea = PrTemp *ITArea
0089
0090          IArS1=ITArea
0091
0092          DigTp1=Digit(1)
0093          PrTemp = Probs(1,DigTp1)
0094          IArea = PrTemp *ITArea
0095
0096          ArHol1 = ArHol1+ITArea*DigTp1
0097          ArHol2 = ArHol2+ITArea*DigTp2
0098          ArHol3 = ArHol3+ITArea*DigTp3
0099          ArHol4 = ArHol4+ITArea*DigTp4
0100          ArHol5 = ArHol5+ITArea*DigTp5
0101
0102      END IF
0103      Intact=Intact+ITArea
0104      Bin = 1
0105      OBinld = .TRUE.
0106      UnInc = .TRUE.
0107      C      DO WHILE (Bin.LE.NBin.AND.UnInc)
0108          24      IF (.NOT.(Bin.LE.NBin.AND.UnInc)) GOTO 240
0109              IF (Digit(Bin).LT.HoMin(Bin)) THEN
0110                  Digit(Bin) = Digit(Bin)+1
0111                  HoDept = HoDept+P(Bin)
0112                  UnInc = .FALSE.
0113              ELSE
0114                  OBinld = .FALSE.
0115                  HoDept = HoDept-Digit(Bin)*P(Bin)
0116                  Digit(Bin)=0
0117                  Bin = Bin + 1
0118              END IF
0119              GOTO 24
0120          240      CONTINUE
0121      C      END DO
0122      END IF
0123      GOTO 2
0124      20      CONTINUE
0125      C      END DO
0126      Nr(1)=DMAX1(0D0,Nr(1)-ArHol1*As/Ac(1))

```

D180-30550-4

```
0127      Nr(2)=DMAX1(0D0,Nr(2)-ArHol2*As/Ac(2))
0128      Nr(3)=DMAX1(0D0,Nr(3)-ArHol3*As/Ac(3))
0129      Nr(4)=DMAX1(0D0,Nr(4)-ArHol4*As/Ac(4))
0130      Nr(5)=DMAX1(0D0,Nr(5)-ArHol5*As/Ac(5))
0131      RETURN
0132      END
```

```

0001
0002      SUBROUTINE ResVel
0003      :   (Vr,V0,RP,RhoP,RhoT,Thick,Theta,PrMat,TarMat)
0004      C       This is the JTCG Residual Velocity Formula
0005      INTEGER PrMat,TarMat
0006      PresAr = 3.14159*RP**2
0007      V0cm = V0/.03281
0008      Weight = 4./3.*3.14159*RP**3*RhoP
0009      V50 = Ballim(RP,RhoP,RhoT,Thick,Theta,PrMat,TarMat)
0010      Q4 = RhoT*Thick*PresAr/(Weight*COS(Theta))
0011      VrSq = AMAX1(0.,V0cm**2-V50**2)
0012      Vr = SQRT(VrSq) / (1.+Q4)
0013      Vr = Vr*.03281
0014      RETURN
0015      END

```

```

0001
0002      FUNCTION BallLim
0003      :      (RP,RhoP,RhoT,Thick,Theta,PrMat,TarMat)
0004      INTEGER PrMat,TarMat
0005      IF (PrMat.EQ.3) THEN
0006          BallLim = 2.45*3281*(Thick/12./2.54)**(-.15)*
0007      :      (Thick/RP/2.)*(RhoT/RhoP)**.64*(1./COS(Theta))**1.01
0008      ELSE
0009      C      This is the JTCG V50 formula
0010          PresAr = 3.14159*RP**2
0011          Weight = 4./3.*3.14159*RP**3*RhoP*32.2
0012          W0 = .0143
0013          IF (PrMat.EQ.1) THEN
0014              IF (TarMat.EQ.2) THEN
0015                  Cbf = 41300.
0016                  Bf = .941
0017                  H = 1.098
0018                  F = -.038
0019              ELSE
0020                  Cbf = 80600.
0021                  Bf = .963
0022                  H = 1.286
0023                  F = -.057
0024              END IF
0025          ELSE
0026              Cbf = 92800
0027              Bf = .972
0028              H = 1.01
0029              F = -.105
0030          END IF
0031          Q8 = RhoP*32.2*Thick*PresAr/Weight
0032          Q11 = RhoP*32.2*Thick*PresAr/W0
0033          BallLim = Cbf*Q8**Bf/COS(Theta)**H*Q11**F
0034      END IF
0035      RETURN
0036      END

```

D180-30550-4

FILE - DEB.VEL

1	0.00	3.21	1.054336E-04	1.368229E-03	7.253182E-04	2.749629E-04
2	3.21	4.00	-4.229632	3.795033	-1.134365	1.135913E-01
3	4.00	4.11	-4.200628E+03	3.110938E+03	-7.679830E+02	6.319804E+01
4	4.11	5.16	-1.622625	7.572647E-01	-8.053452E-02	-4.511672E-04
5	5.16	5.96	2.043598E+01	-1.042294E+01	1.775327	-1.007870E-01
6	5.96	7.18	7.860538	-3.547154	5.318564E-01	-2.631684E-02
7	7.18	8.10	-1.915112E-01	-1.182071E-01	4.134483E-02	-2.761015E-03
8	8.10	8.67	6.661122E+01	-2.432737E+01	2.967511	-1.207315E-01
9	8.67	9.97	-3.037310E+01	1.024428E+01	-1.147075	4.273907E-02
10	9.97	10.25	-4.609121	4.719100E-01	0.0	0.0
11	10.25	10.72	-3.250999E+02	9.131992E+01	-8.543601	2.664182E-01
12	10.72	11.42	3.257151E+02	-8.911450E+01	8.133017	-2.474180E-01
13	11.42	11.56	3.994901	-3.317631E-01	0.0	0.0
14	11.56	12.36	1.701649E+02	-4.206065E+01	3.470595	-9.552701E-02
15	12.36	13.61	4.714685E+01	-1.113553E+01	8.805859E-01	-2.325908E-02
16	13.61	17.00	1.578160E+01	-2.934914	1.820184E-01	-3.763893E-03

D180-30550-4

This page left intentionally blank.

D180-30550-4

APPENDIX D

Source Code for BUMPER Module

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C C
0004 C BUMPER VER 4.0 5/25/87 C
0005 C C
0006 C BOEING AEROSPACE CO. C
0007 C C
0008 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0009 C
0010 C
0011 C
0012 C Bumper Ver 4.0 predicts the probability of no penetration for
0013 C spacecraft subject to man-made orbital debris or meteoroid impact.
0014 C The spacecraft is assumed to be operating in low earth orbit ( approx.
0015 C 500km ). The code accounts for varying impact velocity, impact angle,
0016 C wall configurations, and the effects of spacecraft geometry and
0017 C orientation. It is currently limited to the case of spheres
0018 C ( debris or meteoroids ) impacting conventional aluminum two plate
0019 C structures, with or without multi-layer insulation between the plates.
0020 C
0021 C The code requires 2 files generated by other codes. The first is the
0022 C output file from the GEOMETRY code. This file contains the threat
0023 C information and the element id, pid, and surface area lists. In
0024 C addition the code outputs a list of the exposed elements and their
0025 C impact angles for each threat case.
0026 C
0027 C The second file is the output file from the RESPONSE code. This file
0028 C contains the critical diameter tables for each wall configuration as
0029 C a function of impact velocity and impact angle.
0030 C
0031 C
0032 C The code also produces a Supertab universal file which may be used in
0033 C conjunction with the original Supertab model file to graphically
0034 C display the distribution of the probability of penetration on the
0035 C spacecraft.
0036 C
0037 C The code was developed under the NASA contract 'Integrated Wall Design
0038 C Guide and Penetration Control Plan' by M.A.Wright & A.R.Coronado
0039 C
0040 C
0041 C Common Block Variable list
0042 C
0043 C Scalars
0044 C
0045 C alt = operating altitude , km
0046 C binc = impact angle(beta) increment , deg
0047 C cbeta = cosine of beta
0048 C diam = critical diameter , cm
0049 C etime = spacecraft exposure time , years
0050 C flx = number of impacts per projected area per year of diameter D
0051 C or larger
0052 C it = current threat case
0053 C itype = analysis type , 1- debris, 2-meteoroids
0054 C nb = number of angles in the response array
0055 C nc = number of wall configurations in the response array
0056 C nel = current element number
0057 C nelm = total number of elements
0058 C nr = number of element ranges to sum over
0059 C nt = number of threat cases
0060 C nv = number of velocities in the response array
0061 C vr = impact (relative) velocity , km/sec
0062 C vinc = impact (relative) velocity increment , km/sec
0063 C

```

```

0064      C      Arrays                      D180-30550-4
0065      C
0066      C      area = array containing the value of the surface area for each
0067      C      element, sq-meters
0068      C      exposed = list of the number of exposed elements for each threat
0069      C      angle.
0070      C      geometry = array containing the values of the cosine of the impact
0071      C      angle for each exposed element for each threat angle.
0072      C      id = array containing the values of the element and property id
0073      C      for each element
0074      C      1- id
0075      C      2- pid
0076      C      point = array of the element numbers corresponding to the elements
0077      C      in the geometry array.
0078      C      range = array containing the starting and ending element id for each
0079      C      range to sum over
0080      C      1-starting id
0081      C      2- ending id
0082      C      response = array containing the values of the critical diameter as
0083      C      a function of impact angle and velocity.
0084      C      (vr,beta,pid)
0085      C      threat = array containing threat information
0086      C      1-theta angle, rad
0087      C      2-phi angle, rad
0088      C      3-vr, km/sec
0089      C      4-prob
0090      C
0091      C
0092      C      Main Program Variable List
0093      C
0094      C      Scalars
0095      C
0096      C      answer = user input
0097      C      ae = effective area term, product of the cosine of the impact
0098      C      angle and the probability of the threat occurring
0099      C      fa = product of the threat probability, flux and cosine(beta),
0100      C      impacts/yr/sq-meter
0101      C      prob = probability of threat case i occurring
0102      C      taeff = total effective area, sq-m
0103      C      tnpnp = total probability of no penetration for all elements, %
0104      C      tsum = total sum of the history array
0105      C
0106      C      Arrays
0107      C
0108      C      aeff = array containing the effective area for each range
0109      C      arhis = array containing the running sum of the effective area
0110      C      term for each element
0111      C      history = list containing the running sum of the fa term for each
0112      C      element
0113      C      pnp = array containing the probability of no penetration for each
0114      C      range
0115      C      sum = array containing the sum of the history array for each element
0116      C      range
0117      C
0118      C
0119      C      CHARACTER*80 ANSWER
0120      C
0121      C      INCLUDE 'COMMON2.BLK'
0122      C
0123      C
0124      C      REAL*8 AE,FA,PROB,TAEFF,TPNP,TSUM,
0125      C      1      ARHIS( IEL),HISTORY( IEL),AEFF(50),PNP(50),SUM(50)
0126      C
0127      C
0128      C      Write header to screen, read in screen inputs

```

```

0149      C                               D180-30550-4
0150      C      CALL INPUT
0151      C
0152      C      Read in the GEOMETRY output file
0153      C
0154      C      CALL GEOREAD
0155      C
0156      C      Read in the RESPONSE output file
0157      C
0158      C      CALL RESREAD
0159      C
0160      C      Initialize Tsum and Taeff to 0.0
0161      C
0162      C      TAEFF=0.0D0
0163      C      TSUM=0.0D0
0164      C
0165      C      Initialize History to 0.0
0166      C
0167      C      DO 50 I=1,NELM
0168      C          HISTORY(I)=0.0D0
0169      C      50 CONTINUE
0170      C
0171      C      Initialize Sum, Aeff, and Pnp to 0.0
0172      C
0173      C      DO 70 I=1,50
0174      C          AEFF(I)=0.0D0
0175      C          PNP(I)=0.0D0
0176      C          SUM(I)=0.0D0
0177      C      70 CONTINUE
0178      C
0179      C      Determine the penetrating flux for each element, for each threat
0180      C      angle
0181      C
0182      C      DO 200 I=1,NT
0183      C
0184      C      Set the threat index and get the impact velocity and the threat
0185      C      probability from the threat array
0186      C
0187      C          IT=I
0188      C          VR=THREAT(3,IT)
0189      C          PROB=THREAT(4,IT)
0190      C
0191      C      Evaluate each exposed element
0192      C
0193      C      DO 100 J=1,EXPOSED(I)
0194      C
0195      C      Set the element number
0196      C
0197      C          NEL=POINT(J,I)
0198      C
0199      C      Get the cosine of the impact angle from the Geometry array.
0200      C
0201      C          CBETA=GEOMETRY(J,I)
0202      C
0203      C      Determine the diameter of the sphere that just penetrates the wall
0204      C
0205      C          CALL CRITDIA
0206      C
0207      C      Calculate the flux of the critical diameter using the appropriate
0208      C      flux equation based on the analysis type.
0209      C
0210      C          CALL FLUX
0211      C

```

D180-30550-4

```

0212 C Multiply the flux by the probability of the threat angle and the
0213 C cosine of the impact angle, this determines the number of
0214 C penetrations per year per element surface area for the current
0215 C element and the current threat angle.
0216 C
0217 C      FA=FLX*PROB*CBETA
0218 C
0219 C Store the running sum of the FA term for each element in the History
0220 C array. This value represents the total number of penetrations for
0221 C a given element per year per element surface area.
0222 C
0223 C      HISTORY(NEL)=HISTORY(NEL)+FA
0224 C
0225 C Calculate the effective area term
0226 C
0227 C      AE=PROB*CBETA
0228 C
0229 C Store the running sum of the effective area term in the arhis array
0230 C
0231 C      ARHIS(NEL)=ARHIS(NEL)+AE
0232 C
0233 C Next element
0234 C
0235 100 CONTINUE
0236 C
0237 C Write message to the screen indicating current threat has been
0238 C evaluated
0239 C
0240 C      WRITE ( 6,150 ) IT
0241 150 FORMAT ( 1X,'THREAT CASE ',I4,1X,'COMPLETED' )
0242 C
0243 C Next threat
0244 C
0245 200 CONTINUE
0246 C
0247 C Multiply each term in the HISTORY and ARHIS arrays by the appropriate
0248 C element surface area.
0249 C
0250 C      DO 250 I=1,NELM
0251 C          HISTORY(I)=HISTORY(I)*AREA(I)
0252 C          ARHIS(I)=ARHIS(I)*AREA(I)
0253 250 CONTINUE
0254 C
0255 C Sum up the HISTORY and ARHIS arrays by components.
0256 C
0257 C      IC=1
0258 C
0259 C      DO 310 I=1,NELM
0260 C
0261 C          TAEFF=TAEFF+ARHIS(I)
0262 C          TSUM=TSUM+HISTORY(I)
0263 C
0264 300 IS=RANGE(1,IC)
0265 IE=RANGE(2,IC)
0266 C
0267 C      IF ( IC.GT.NR ) GO TO 310
0268 C      IF ( IS.GT.ID(1,I) ) GO TO 310
0269 C      IF ( IE.LT.ID(1,I) ) THEN
0270 C          IC=IC+1
0271 C          GO TO 300
0272 C      END IF
0273 C
0274 C      AEFF(IC)=AEFF(IC)+ARHIS(I)

```

```

0275      SUM(IC)=SUM(IC)+HISTORY(I)
0276      C
0277      310 CONTINUE
0278      C
0279      C Calculate the probability of no penetration for each range using
0280      C a Poission model
0281      C
0282      DO 320 I=1,NR
0283      PNP(I)=(DEXP(-SUM(I)*ETIME))*100.D0
0284      320 CONTINUE
0285      C
0286      C Calculate the total PNP
0287      C
0288      TPNP=(DEXP(-TSUM*ETIME))*100.0D0
0289      C
0290      C Write the probability value to the screen and the summary file.
0291      C
0292      WRITE ( 6,390 )
0293      WRITE ( 10,390 )
0294      390 FORMAT ( /1X,'RANGE',2X,'STARTING EID',2X,'ENDING EID',3X,
0295      1 'PNP %',6X,'Aeff sq-m' )
0296      DO 400 I=1,NR
0297      WRITE ( 6,420 ) I,RANGE(1,I),RANGE(2,I),PNP(I),AEFF(I)
0298      WRITE ( 10,420 ) I,RANGE(1,I),RANGE(2,I),PNP(I),AEFF(I)
0299      400 CONTINUE
0300      420 FORMAT ( 2X,I4,5X,I8,3X,I8,3X,F10.5,2X,F10.5 )
0301      C
0302      C Write out totals to screen and save file
0303      C
0304      WRITE ( 6,430 ) TPNP
0305      WRITE ( 10,430 ) TPNP
0306      430 FORMAT ( /1X,'TOTAL PRROBABILITY OF NO PENETRATION (%) = ',F12.5 )
0307      C
0308      WRITE ( 6,440 ) TAEFF
0309      WRITE ( 10,440 ) TAEFF
0310      440 FORMAT ( /1X,'TOTAL EFFECTIVE AREA (SQ-M) = ',F12.5 )
0311      C
0312      C
0313      C Determine if a Supertab output file is to be created
0314      C
0315      WRITE ( 6,500 )
0316      500 FORMAT (/1X,'CREATE A SUPERTAB INPUT FILE FOR CONTOUR PLOTS ?',
0317      1 1X,'<CR=YES> >',$)
0318      C
0319      READ ( 5,510 ) ANSWER
0320      510 FORMAT ( A )
0321      C
0322      IF ( ANSWER(1:1).EQ.'Y' .OR. ANSWER(1:1).EQ.' ' ) THEN
0323      CALL SUPER (HISTORY)
0324      END IF
0325      C
0326      C Close summary file
0327      C
0328      CLOSE ( UNIT=10,STATUS='KEEP' )
0329      C
0330      C Finished
0331      C
0332      END

```

```

0001      C                               D180-30550-4
0002      C
0003      C
0004      C      SUBROUTINE INPUT
0005      C
0006      C
0007      C
0008      C
0009      C      Input writes the program header to the screen and reads in the
0010      C      summary output filename. It also determines the analysis type, the
0011      C      spacecraft exposure time, operating altitude and the element id sum
0012      C      ranges.
0013      C
0014      C
0015      C
0016      C      note: for variables contained in the common block refer to the main
0017      C      listing for definition
0018      C
0019      C      Variable list
0020      C
0021      C      answer = character string representing user input
0022      C      sfile = summary output filename
0023      C
0024      C
0025      C
0026      C      INCLUDE 'COMMON2.BLK'
0049      C
0050      C      CHARACTER*80 ANSWER,SFILE
0051      C
0052      C      Write header to screen and summary file
0053      C
0054      C      WRITE ( 6,10 )
0055      C      10 FORMAT (/1X,'*****',//1X,3X,'BUMPER VER 4.0',
0056      C      1      //1X,'Last Update 5/25/87',//1X,'*****')
0057      C
0058      C      Read in summary output filename, set default to sum.dat
0059      C
0060      C      15 WRITE ( 6,20 )
0061      C      20 FORMAT ( /1X,'SUMMARY OUTPUT FILENAME (CR=BUMPER.SUM)>',$)
0062      C      READ ( 5,30 )SFILE
0063      C      30 FORMAT (A)
0064      C
0065      C      IF ( SFILE(1:1).EQ.' ' ) SFILE='BUMPER.SUM'
0066      C
0067      C      Open sfile
0068      C
0069      C      OPEN ( UNIT=10,FILE=SFILE,STATUS='NEW',IOSTAT=IER,ERR=40 )
0070      C
0071      C      GO TO 70
0072      C
0073      C      Error control
0074      C
0075      C      40 IF ( IER.EQ.2013 ) THEN
0076      C      WRITE ( 6,50 )
0077      C      50 FORMAT ( /1X,'FILE ALREADY EXISTS OK TO OVERWRITE (CR=YES,$)>')
0078      C      READ ( 5,30 ) ANSWER
0079      C
0080      C      IF ( ANSWER(1:1).EQ.'Y' .OR. ANSWER(1:1).EQ.' ' ) THEN
0081      C      OPEN ( UNIT=10,FILE=SFILE,STATUS='UNKNOWN',IOSTAT=IER,
0082      C      1      ERR=40)
0083      C      REWIND 10
0084      C      ELSE
0085      C      GO TO 15

```

D180-30550-4

```

0086         END IF
0087     ELSE
0088         WRITE ( 6,60 )
0089     60     FORMAT (/1X,'UNABLE TO OPEN FILE ' )
0090         GO TO 15
0091     END IF
0092 C
0093 C Determine analysis type, set default to 1 (debris)
0094 C
0095     70 WRITE ( 6,80 )
0096     80 FORMAT (/1X,'ANALYSIS TYPE ?',/,2X,'1-DEBRIS <CR> ',/,2X,
0097     1      '2-METEOROIDS',/,1X,'ANSWER 1 OR 2 >',$)
0098 C
0099     READ ( 5,30 ) ANSWER
0100 C
0101     IF ( ANSWER(1:1).EQ.' ' ) THEN
0102         ITYPE=1
0103     ELSE
0104         READ ( ANSWER(1:80),90,ERR=70 ) ITYPE
0105     90     FORMAT ( BN,I4 )
0106     END IF
0107 C
0108 C Check that input was correct
0109 C
0110     IF ( ITYPE.EQ.1 .OR. ITYPE.EQ.2 ) THEN
0111         CONTINUE
0112     ELSE
0113         WRITE ( 6,100 )
0114     100     FORMAT ( /1X,'INCORRECT INPUT' )
0115         GO TO 70
0116     END IF
0117 C
0118 C Determine the spacecraft exposure time , set default to 10 years
0119 C
0120     105 WRITE ( 6,110 )
0121     110 FORMAT (/1X,'SPACE STATION EXPOSURE TIME (YEARS) <CR=10.0> : ',,$)
0122     READ ( 5,30 ) ANSWER
0123 C
0124     IF ( ANSWER(1:1).EQ.' ' ) ANSWER='10.0'
0125 C
0126     READ ( ANSWER(1:80),120,ERR=105 ) ETIME
0127     120 FORMAT ( BN,D20.0 )
0128 C
0129 C Write analysis type and etime to summary file
0130 C
0131     WRITE ( 10,10 )
0132 C
0133     IF ( ITYPE.EQ.1 ) THEN
0134         WRITE ( 10,130 )
0135     130     FORMAT ( /1X,'MAN-MADE ORBITAL DEBRIS ANALYSIS')
0136     ELSE
0137         WRITE ( 10,140 )
0138     140     FORMAT ( 1X,'METEOROID ANALYSIS' )
0139     END IF
0140 C
0141     WRITE ( 10,150 ) ETIME
0142     150 FORMAT ( 1X,'SPACECRAFT EXPOSURE TIME (YEARS) =' ,F7.2 )
0143 C
0144 C Read in operating altitude , set default to 500 km
0145 C
0146     160 WRITE ( 6,170 )
0147     170 FORMAT ( /1X,'OPERATING ALTITUDE ( 400-500 km ) <CR=500> : ',,$)
0148     READ ( 5,30 ) ANSWER

```

D180-30550-4

```

0149 C
0150 IF ( ANSWER(1:1).EQ.' ' ) THEN
0151 ALT=500.0D0
0152 ELSE
0153 READ ( ANSWER(1:80),180,ERR=160 ) ALT
0154 180 FORMAT ( BN,D20.0 )
0155 END IF
0156 C
0157 C Check that altitude is within range
0158 C
0159 IF ( ALT.LT.350.0 .OR. ALT.GT.550.0 ) THEN
0160 WRITE ( 6,190 )
0161 190 FORMAT ( 1X,'---ERROR--- Altitude outside of range' )
0162 GO TO 160
0163 END IF
0164 C
0165 C Write altitude to output file
0166 C
0167 WRITE ( 10,200 ) ALT
0168 200 FORMAT ( 1X,'OPERATING ALTITUDE (km) = ',F7.2 )
0169 C
0170 C Read in element ranges to sum over
0171 C
0172 IC=0
0173 C
0174 WRITE ( 6,250 )
0175 250 FORMAT ( /1X,'THE PROBABILITY OF NO PENETRATION WILL BE ',
0176 1 'CALCULATED FOR SPECIFIC RANGES',/1X,'OF ELEMENT IDS ',
0177 2 'INPUT THE STARTING AND ENDING ELEMENT ID FOR ',
0178 3 'EACH RANGE',/1X,'ENTER D <CR> OR <CR> WHEN DONE')
0179 C
0180 270 IC=IC+1
0181 275 WRITE ( 6,280 ) IC
0182 280 FORMAT ( /1X,'RANGE',I4 )
0183 C
0184 285 WRITE ( 6,290 )
0185 290 FORMAT ( 1X,'STARTING ELEMENT ID : ',)$)
0186 READ ( 5,30 ) ANSWER
0187 C
0188 IF ( ANSWER(1:1).EQ.' ' .OR. ANSWER(1:1).EQ.'D' ) GO TO 500
0189 C
0190 READ ( ANSWER(1:80),300,ERR=285 ) RANGE(1,IC)
0191 300 FORMAT ( BN,I12 )
0192 C
0193 305 WRITE ( 6,310 )
0194 310 FORMAT ( 1X,'ENDING ELEMENT ID : ',)$)
0195 READ ( 5,30 ) ANSWER
0196 C
0197 IF ( ANSWER(1:1).EQ.' ' .OR. ANSWER(1:1).EQ.'D' ) GO TO 305
0198 C
0199 READ ( ANSWER(1:80),300,ERR=305 ) RANGE(2,IC)
0200 C
0201 C Check that ending id > starting id
0202 C
0203 IF ( RANGE(1,IC).GT.RANGE(2,IC) ) THEN
0204 WRITE ( 6,320 )
0205 320 FORMAT ( 1X,'---ERROR--- Staring ID greater then Ending ID')
0206 GO TO 275
0207 END IF
0208 C
0209 C Next Range
0210 C
0211 GO TO 270

```

D180-30550-4

```

0212 C
0213 C Check that values were input
0214 C
0215 500 CONTINUE
0216 IF ( RANGE(1,1).EQ.0.0 .AND. RANGE(2,1).EQ.0.0 ) THEN
0217 WRITE ( 6,330 )
0218 330 FORMAT ( 1X,'---ERROR--- No Range Values Input' )
0219 IC=0
0220 GO TO 270
0221 END IF
0222 C
0223 C Set the number of ranges equal to the number read in
0224 C
0225 NR=IC-1
0226 C
0227 C Finished
0228 C
0229 C RETURN
0230 C
0231 C END

```

```

0001      C                                     D180-30550-4
0002      C
0003      C
0004      C          SUBROUTINE GEOREAD
0005      C
0006      C
0007      C
0008      C
0009      C      Georead reads in the output file from the GEOMETRY code. This file
0010      C      contains the global threat and element data as well as the list of
0011      C      exposed elements and their impact angles for each threat case.
0012      C
0013      C
0014      C
0015      C      note: for variables contained in the common block refer to main
0016      C      listing for definition
0017      C
0018      C
0019      C      Variable List
0020      C
0021      C      answer = character string representing user input
0022      C      gfile = geometry output filename
0023      C      itf = analysis type contained in the
0024      C
0025      C
0026      C          INCLUDE 'COMMON2.BLK'
0049      C
0050      C          CHARACTER*80 ANSWER,GFILE
0051      C
0052      C          INTEGER*2 ITF
0053      C
0054      C      Read in the GEOMETRY output filename, set the default to geom.dat
0055      C
0056      C      10 WRITE ( 6,20 )
0057      C      20 FORMAT(/1X,'GEOMETRY OUTPUT FILENAME ? <CR=STATION.GEM> >','$)
0058      C      READ ( 5,30 ) GFILE
0059      C      30 FORMAT (A)
0060      C
0061      C          IF ( GFILE(1:1).EQ.' ' ) GFILE='STATION.GEM'
0062      C
0063      C      Open the file
0064      C
0065      C          OPEN (UNIT=2,FILE=GFILE,STATUS='OLD',FORM='UNFORMATTED',ERR=40 )
0066      C
0067      C          GO TO 60
0068      C
0069      C      Error control
0070      C
0071      C      40 WRITE ( 6,50 )
0072      C      50 FORMAT ( /1X,'UNABLE TO OPEN FILE ' )
0073      C      GO TO 10
0074      C
0075      C      Read in the analysis type,the number of threat cases, and the
0076      C      number of elements
0077      C
0078      C      60 READ (2) ITF,NT,NELM
0079      C
0080      C      Check that the analysis type in the file matches the type input
0081      C      by user.
0082      C
0083      C          IF ( ITF.NE.ITYPE ) THEN
0084      C              IF ( ITYPE.EQ.1 ) THEN
0085      C                  WRITE ( 6,70 )

```

D180-30550-4

```

0086      70      FORMAT ( /1X,'DEBRIS ANALYSIS SPECIFIED BUT FILE IS FOR ',
0087      1          'METEORIDS')
0088      ELSE
0089          WRITE ( 6,80 )
0090      80      FORMAT ( /1X,'METEOROID ANALYSIS SPECIFIED BUT FILE IS ',
0091      1          'FOR DEBRIS' )
0092      END IF
0093      C
0094          WRITE ( 6,90 )
0095      90      FORMAT ( /1X,'DO YOU WISH TO CONTINUE (CR=NO) >',$)
0096      READ ( 5,30 ) ANSWER
0097      C
0098          IF ( ANSWER(1:1).EQ.'Y' ) THEN
0099              GO TO 10
0100          ELSE
0101              STOP
0102          END IF
0103      C
0104      END IF
0105      C
0106      C Check that the number of threats and the number of elements are less
0107      C then the maximum allowed
0108      C
0109          IF ( NT.GT.ITH ) THEN
0110              WRITE ( 6,100 )
0111      100      FORMAT ( /1X,'NUMBER OF THREATS IS GREATER THEN MAX ALLOWED')
0112              WRITE ( 6,105 )
0113      105      FORMAT ( 1X,'ARRAY SIZE MUST BE INCREASED & CODE RECOMPILED')
0114              STOP
0115          END IF
0116      C
0117          IF ( NELM.GT.IEL ) THEN
0118              WRITE ( 6,110 )
0119      110      FORMAT ( /1X,'NUMBER OF ELEMENTS IS GREATER THEN MAX ALLOWED')
0120              WRITE ( 6,105 )
0121              STOP
0122          END IF
0123      C
0124      C Initialize the arrays to 0.0
0125      C
0126          DO 150 I=1,NT
0127              THREAT(3,I)=0.0
0128              THREAT(4,I)=0.0
0129              EXPOSED(I)=0
0130              DO 140 J=1,NELM
0131                  GEOMETRY(J,I)=0.0
0132                  ID(1,J)=0
0133                  ID(2,J)=0
0134                  POINT(J,I)=0
0135      140      CONTINUE
0136      150 CONTINUE
0137      C
0138      C Read in the Threat data
0139      C
0140          DO 175 I=1,NT
0141              READ (2) (THREAT(J,I),J=1,4)
0142      175 CONTINUE
0143      C
0144      C Read in the element id, and property id storing them in the ID
0145      C array.
0146      C
0147          DO 200 I=1,NELM
0148              READ (2) (ID(J,I),J=1,2)

```

D180-30550-4

```

0149      200 CONTINUE
0150      C
0151      C   Read in the element's surface area storing it in the AREA array.
0152      C
0153          DO 250 I=1,NELM
0154              READ (2) AREA(I)
0155      250 CONTINUE
0156      C
0157      C   Read in the geometry data for the exposed elements
0158      C
0159          DO 500 I=1,NT
0160      C
0161      C   Read in the threat case and the number of exposed elements
0162      C
0163          READ (2) IT,EXPOSED(I)
0164      C
0165      C   Loop thru the exposed elements
0166      C
0167          DO 400 J=1,EXPOSED(I)
0168      C
0169      C   Read in the element number (storing in the POINT array), and the
0170      C   cosine of the impact angle (storing in the GEOMETRY array).
0171      C
0172          READ (2) POINT(J,I),GEOMETRY(J,I)
0173      C
0174      400 CONTINUE
0175      C
0176      500 CONTINUE
0177      C
0178      C   Write gfile to summary file
0179      C
0180          WRITE ( 10,600 )GFILE
0181      600 FORMAT ( 1X,'GEOMETRY OUTPUT FILE = ',A )
0182      C
0183      C   Close file
0184      C
0185          CLOSE ( UNIT=2,STATUS='KEEP' )
0186      C
0187      RETURN
0188      C
0189      END

```

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE RESREAD
0005 C
0006 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007 C
0008 C
0009 C Resread reads in the output from the RESPONSE code. This output
0010 C consists of the critical diameter data as a function of property
0011 C id, impact angle, and impact velocity.
0012 C
0013 C
0014 C note: for variables contained in the common block refer to the main
0015 C listing for definition.
0016 C
0017 C
0018 C Variable list
0019 C
0020 C answer = character string represnting user input
0021 C itf = analysis type for rfile
0022 C rfile = response output filename
0023 C
0024 C
0025 CHARACTER*80 ANSWER,RFILE
0026 C
0027 INTEGER*2 ITF
0028 C
0029 INCLUDE 'COMMON2.BLK'
0052 C
0053 C Read in the RESPONSE output filename , set default to resp.dat
0054 C
0055 10 WRITE ( 6,20 )
0056 20 FORMAT(/1X,'RESPONSE OUTPUT FILENAME ? <CR=STATION.RSP> >',$)
0057 READ ( 5,30 ) RFILE
0058 30 FORMAT (A)
0059 C
0060 IF ( RFILE(1:1).EQ.' ' ) RFILE='STATION.RSP'
0061 C
0062 C Open the file
0063 C
0064 OPEN ( UNIT=2,FILE=RFILE,STATUS='OLD',FORM='UNFORMATTED',ERR=40 )
0065 C
0066 GO TO 60
0067 C
0068 C Error control on open
0069 C
0070 40 WRITE ( 6,50 )
0071 50 FORMAT ( /1X,'UNABLE TO OPEN FILE' )
0072 GO TO 10
0073 C
0074 C Read in the analysis type and the number of property cases.
0075 C
0076 60 READ (2) ITF,NC
0077 C
0078 C Check that the response file is the correct analysis type
0079 C
0080 IF ( ITF.NE.ITYPE ) THEN
0081 IF ( ITYPE.EQ.1 ) THEN
0082 WRITE ( 6,70 )
0083 70 FORMAT ( /1X,'DEBRIS ANALYSIS SPECIFIED BUT RESPONSE FILE',
0084 1 ' IS FOR METEOROIDS ' )
0085 ELSE

```

```

0086          WRITE ( 6,80 )
0087      80      FORMAT ( /1X, 'METEOROID ANALYSIS SPECIFIED BUT RESPONSE',
0088          1      ' FILE IS FOR DEBRIS' )
0089          END IF
0090      C
0091          WRITE ( 6,90 )
0092      90      FORMAT ( /1X, 'DO YOU WISH TO CONTINUE (CR=NO) >', $)
0093          READ ( 5,30 ) ANSWER
0094      C
0095          IF ( ANSWER(1:1).EQ.'Y' ) THEN
0096              GO TO 10
0097          ELSE
0098              STOP
0099          END IF
0100      C
0101          END IF
0102      C
0103      C Read in the impact angle information
0104      C
0105          READ (2) NB,BINC
0106      C
0107      C Read in the impact velocity information
0108      C
0109          READ (2) NV,VINC
0110      C
0111      C Initialize RESPONSE to 0.0
0112      C
0113          DO 200 I=1,NC
0114              DO 150 J=1,NB
0115                  DO 100 K=1,NV
0116                      RESPONSE ( K,J,I ) = 0.0
0117          100      CONTINUE
0118          150      CONTINUE
0119          200      CONTINUE
0120      C
0121      C Read in the critical diameter data
0122      C
0123      C Loop thru the property id's
0124      C
0125          DO 400 I=1,NC
0126      C
0127      C Loop thru the impact angles
0128      C
0129          DO 300 J=1,NB
0130      C
0131      C Loop thru the impact velocities
0132      C
0133          DO 250 K=1,NV
0134      C
0135      C Store the critical diameter in response
0136      C
0137          READ (2) RESPONSE(K,J,I)
0138      250      CONTINUE
0139      300      CONTINUE
0140      400      CONTINUE
0141      C
0142      C Close the file and return
0143      C
0144          CLOSE ( UNIT=2,STATUS='KEEP' )
0145      C
0146      C Write Rfile to summary file
0147      C
0148          WRITE ( 10,500 )RFILE

```

D180-30550-4

```
0149      500 FORMAT(1X, 'RESPONSE OUTPUT FILE = ', A )
0150      C
0151      RETURN
0152      C
0153      END
```

```

0001      C                               D180-30550-4
0002      C
0003      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0004      C
0005      SUBROUTINE CRITDIA
0006      C
0007      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008      C
0009      C
0010      C Critdia determines the diameter of the sphere that just penetrates
0011      C the given wall configuration at the given impact angle and velocity.
0012      C The subroutine performs a linear interpolation using the appropriate
0013      C values from the Response array to estimate the critical diameter.
0014      C
0015      C note: for variables contained in the common block refer to the main
0016      C listing for definition
0017      C
0018      C
0019      C Variable list
0020      C
0021      C beta = impact angle , measured from the normal , deg
0022      C b1 = the impact angle nearest to the actual impact angle
0023      C in the Response array , but less than the actual , deg
0024      C ib1 = location of b1 in the Response array
0025      C ib2 = " " b2 " " " "
0026      C iv1 = " " v1 " " " "
0027      C iv2 = " " v2 " " " "
0028      C r1 = intermediate variable
0029      C r2 = " "
0030      C r11 = value in Response array at location iv1,ib1,pid
0031      C r12 = " " " " " " iv1,ib2,pid
0032      C r21 = " " " " " " iv2,ib1,pid
0033      C r22 = " " " " " " iv2,ib2,pid
0034      C v1 = impact velocity nearest the actual impact velocity in the
0035      C Response array , but still less than the actual
0036      C
0037      C
0038      C
0039      C
0040      C INTEGER*2 IB1,IB2,IV1,IV2,PID
0041      C
0042      C INCLUDE 'COMMON2.BLK'
0043      C
0044      C
0045      C PARAMETER (PI=3.1415926536)
0046      C
0047      C
0048      C Determine the location of the nearest velocity to the actual velocity
0049      C in the Response array, but still less than the actual
0050      C
0051      C IV1=VR/VINC
0052      C
0053      C Check that the location is inside the array
0054      C
0055      C IF ( IV1.LT.1 .OR. IV1.GT.NV ) THEN
0056      C WRITE ( 6,10 ) VR
0057      C 10 FORMAT ( /1X,'THE RELATIVE VELOCITY (VR) IS OUTSIDE OF THE',
0058      C 1 ' RESPONSE ARRAY BOUNDS VR (KM/SEC) = ',E12.5)
0059      C STOP
0060      C END IF
0061      C
0062      C Set the location of the velocity just greater than the actual velocity
0063      C
0064      C IV2=IV1+1
0065      C

```

D180-30550-4

```

0086 C Check that the location is inside the array
0087 C
0088 IF ( IV2.GT.NV ) IV2=IV1-1
0089 C
0090 C Calculate the value of the velocity at location iv1
0091 C
0092 V1=IV1*VINC
0093 C
0094 C Determine the impact angle in deg
0095 C
0096 BETA=ACOS(CBETA)*180.0/PI
0097 C
0098 C Determine the location of the nearest impact angle to the actual
0099 C impact angle in the Response array, but still less than the actual
0100 C
0101 IB1=BETA/BINC+1
0102 C
0103 C Check that the location is inside the array
0104 C
0105 IF ( IB1.LT.1 .OR. IB1.GT.NB ) THEN
0106 WRITE ( 6,20 )BETA
0107 20 FORMAT ( /1X,'IMPACT ANGLE (BETA) IS OUTSIDE THE BOUNDS OF',
0108 1 ' THE RESPONSE ARRAY BETA (DEG) = ',E12.5)
0109 STOP
0110 END IF
0111 C
0112 C Set the location of the impact angle in the Response array that is
0113 C just greater than the actual
0114 C
0115 IB2=IB1+1
0116 C
0117 C Check that the location is inside the array
0118 C
0119 IF ( IB2.GT.NB ) IB2=IB1-1
0120 C
0121 C Calculate the value of the impact angle at location ib1 in the Response
0122 C array
0123 C
0124 B1=(IB1-1)*BINC
0125 C
0126 C Determine the property id
0127 C
0128 PID=ID(2,NEL)
0129 C
0130 C Check that pid is within bounds of the response array
0131 C
0132 IF ( PID.GT.NC ) THEN
0133 WRITE ( 6,30 ) PID
0134 30 FORMAT ( /1X,'NO DATA EXISTS FOR PORPERTY ID ',I4,'IN THE ',
0135 1 'RESPONSE FILE' )
0136 STOP
0137 END IF
0138 C
0139 C Get the four values that surround the actual value in the Response
0140 C array
0141 C
0142 R11=RESPONSE(IV1,IB1,PID)
0143 R12=RESPONSE(IV1,IB2,PID)
0144 R21=RESPONSE(IV2,IB1,PID)
0145 R22=RESPONSE(IV2,IB2,PID)
0146 C
0147 C Using linear interpolation, estimate the critical diameter
0148 C

```

D180-30550-4

```
0149      R1=(R12-R11)*( (BETA-B1)/BINC)+R11
0150      R2=(R22-R21)*( (BETA-B1)/BINC)+R21
0151      C
0152      DIAM=(R2-R1)*( (VR-V1)/VINC)+R1
0153      C
0154      C Finished , return
0155      C
0156      RETURN
0157      C
0158      END
```

D180-30550-4

```

0001 C
0002 C
0003 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0004 C
0005 SUBROUTINE FLUX
0006 C
0007 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008 C
0009 C
0010 C Flux calculates the meteoroid or debris flux for the given critical
0011 C diameter based on analysis type.
0012 C
0013 C
0014 C note: for variables contained in the common block referr to the main
0015 C listing for definition
0016 C
0017 C Variable List
0018 C
0019 C ddiam = diam in double precision , cm
0020 C ge = gravity focusing factor
0021 C intercept = intercept of the flux equation
0022 C mass = critical meteoroid mass, g
0023 C mden = meteoroid density, g/cc
0024 C re = earth's radius (including 100km atmosphere), km
0025 C slope = slope of the flux equation
0026 C
0027 C
0028 C INCLUDE 'COMMON2.BLK'
0051 C
0052 C REAL*8 DDIAM,GE,INTERCEPT,LD,MASS,MDEN,PI,RE,SLOPE
0053 C
0054 C PARAMETER (PI=3.141592653589793238D0)
0055 C
0056 C Set mden
0057 C
0058 C
0059 C MDEN=0.50D0
0060 C
0061 C Calculate the focusinng factor, equation
0062 C is from JSC-30000
0063 C
0064 C RE=6478.0D0
0065 C GE=0.568D0+0.432D0*(RE/(RE+ALT))
0066 C
0067 C Convert diam to double precision
0068 C
0069 C DDIAM=DIAM
0070 C
0071 C Calculate the flux
0072 C
0073 C IF ( ITYPE.EQ.1 ) THEN
0074 C
0075 C For debris use JSC-20001, use stated equations for diameters
0076 C less then 1 cm , for those greater use third order fit of the
0077 C curve for region up to 5 cm .
0078 C
0079 C The log of the flux varies linearly between 400 and 500 km according
0080 C to D Kessler of JSC.
0081 C
0082 C LD=DLOG10(DDIAM)
0083 C IF ( DIAM.LT.5.0 ) THEN
0084 C IF ( DIAM.LT.1.0 ) THEN
0085 C SLOPE=-0.0010D0*ALT-2.0200D0

```

D180-30550-4

```

0086             ELSE
0087                 SLOPE=-0.0022D0*ALT-0.1400D0
0088             END IF
0089             INTERCEPT=+0.0036D0*ALT-7.26D0
0090             FLX=10.0D0**(SLOPE*LD+INTERCEPT)
0091         ELSE
0092             WRITE ( 6,100 )
0093     100         FORMAT ( /1X,'DIAMETER IS GREATER THAN 5 CM LIMIT')
0094             STOP
0095         END IF
0096     C
0097     C   Correct Flux for differance in Boeing and Nasa definetion
0098     C
0099         FLX=FLX*4.0D0
0100     C
0101         ELSE
0102     C
0103     C   For meteoroids use JSC-3000,  E-06g < mass < 1g
0104     C
0105         MASS=PI*(DDIAM**3)/6.0D0*MDEN
0106         FLX=10.0D0**(-14.37D0-1.213D0*DLOG10(MASS))
0107     C
0108     C   Account for earth shielding and gravity focusing , also convert to
0109     C   number of impacts per sq-m per year
0110     C
0111         FLX=FLX*GE*3.15576D07
0112     C
0113         END IF
0114     C
0115         RETURN
0116     C
0117         END

```

D180-30550-4

```

0001
0002
0003 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0004 C
0005 SUBROUTINE SUPER (HISTORY)
0006 C
0007 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008 C
0009 C
0010 C Super writes out the Supertab universal file that contains the
0011 C probability of penetration per surface area data.
0012 C
0013 C
0014 C Variable list
0015 C
0016 C answer = character string representing user input
0017 C ids = dataset id , from Supertab manual
0018 C ilt = leading and trailing -1
0019 C i61 = field value from Supertab manual
0020 C i62 = " " " " "
0021 C i63 = " " " " "
0022 C i64 = " " " " "
0023 C i65 = " " " " "
0024 C i66 = " " " " "
0025 C i71 = " " " " "
0026 C i72 = " " " " "
0027 C i73 = " " " " "
0028 C i91 = " " " " "
0029 C i92 = " " " " "
0030 C ofile = output filename
0031 C ppen = probability of penetration per surface area , %/sq-meter
0032 C r81 = field value from Supertab manual
0033 C
0034 C
0035 C INCLUDE 'COMMON2.BLK'
0058 C
0059 C CHARACTER*80 ANSWER,OFIL
0060 C
0061 C REAL*8 HISTORY(IEL)
0062 C
0063 C Read in output filename, set default to out.uni
0064 C
0065 C 10 WRITE ( 6,20 )
0066 C 20 FORMAT(/1X,'SUPERTAB INPUT FILENAME ? <CR=OUT.UNI>',$)
0067 C READ ( 5,30 ) OFIL
0068 C 30 FORMAT( A )
0069 C
0070 C IF ( OFIL(1:1).EQ.' ' ) OFIL='OUT.UNI'
0071 C
0072 C Open the output file
0073 C
0074 C OPEN ( UNIT=2,FILE=OFIL,STATUS='NEW',IOSTAT=IER,ERR=40 )
0075 C
0076 C GO TO 70
0077 C
0078 C Error control on open
0079 C
0080 C 40 IF ( IER.EQ.2013 ) THEN
0081 C WRITE ( 6,50 )
0082 C 50 FORMAT ( /1X,'FILE ALREADY EXISTS OK TO OVERWRITE (CR=YES) >')
0083 C READ ( 5,30 ) ANSWER
0084 C
0085 C IF ( ANSWER(1:1).EQ.'Y' .OR. ANSWER(1:1).EQ.' ' ) THEN

```

D180-30550-4

```

0086          OPEN ( UNIT=2,FILE=OFILE,STATUS='UNKNOWN',IOSTAT=IER,
0087      1          ERR=40 )
0088          REWIND 2
0089          END IF
0090          ELSE
0091          WRITE ( 6,60 )
0092      60      FORMAT ( /1X,'UNABLE TO OPEN FILE ' )
0093          GO TO 10
0094          END IF
0095      C
0096      C Write the leading -1
0097      C
0098      70      ILT=-1
0099          WRITE ( 2,100 ) ILT
0100      100     FORMAT (I6)
0101      C
0102      C Write the dataset id
0103      C
0104          IDS=56
0105          WRITE ( 2,110 ) IDS
0106      110     FORMAT (I6)
0107      C
0108      C Write the id information
0109      C
0110          IF ( ITYPE.EQ.1 ) THEN
0111              WRITE ( 2,120 )
0112          ELSE
0113              WRITE ( 2,125 )
0114          END IF
0115      C
0116      120     FORMAT ( 1X,'MAN-MADE ORBITAL DEBRIS ANALYSIS')
0117      125     FORMAT ( 1X,'METEOROID ANALYSIS')
0118      C
0119          WRITE ( 2,130 )
0120      130     FORMAT(1X,'PROBABILITY OF PENETRATION (%) PER SQ-METER')
0121      C
0122          DO 150 I=1,3
0123              WRITE ( 2,140 )
0124      140      FORMAT ('NONE')
0125      150     CONTINUE
0126      C
0127      C Write out record 6
0128      C
0129      C Set the 6 field values
0130      C
0131          I61=1
0132          I62=0
0133          I63=1
0134          I64=1
0135          I65=2
0136          I66=1
0137      C
0138          WRITE ( 2,160 ) I61,I62,I63,I64,I65,I66
0139      160     FORMAT ( 6I10 )
0140      C
0141      C Write record 7
0142      C
0143      C Set the 3 field values
0144      C
0145          I71=1
0146          I72=1
0147          I73=1
0148      C

```

```

0149      WRITE ( 2,170 ) I71,I72,I73
0150 170 FORMAT ( 3I10 )
0151 C
0152 C Write out record 8
0153 C
0154 C Set the single field value
0155 C
0156      R81=0.0
0157 C
0158      WRITE ( 2,180 ) R81
0159 180 FORMAT (E13.5)
0160 C
0161 C Write out record 9 & 10 for each element
0162 C
0163      DO 250 I=1,NELM
0164 C
0165 C Calculate the probability of penetration per surface area
0166 C
0167      PPEN=HISTORY(I)*100.0 /AREA(I)
0168 C
0169 C If the value of ppen is < .000001 no need to write data out
0170 C
0171      IF ( PPEN.LT.1.0E-06 ) GO TO 250
0172 C
0173 C Write record 9
0174 C
0175 C Set the 2 field values
0176 C
0177      I91=ID(1,I)
0178      I92=1
0179 C
0180      WRITE ( 2,190 ) I91,I92
0181 190 FORMAT ( 2I10 )
0182 C
0183 C Write record 10
0184 C
0185      WRITE ( 2,200 ) PPEN
0186 200 FORMAT ( E13.5 )
0187 C
0188 C Next element
0189 C
0190 250 CONTINUE
0191 C
0192 C Write out trailing -1
0193 C
0194      WRITE ( 2,260 ) ILT
0195 260 FORMAT ( I6 )
0196 C
0197 C Close the file
0198 C
0199      CLOSE ( UNIT=2 , STATUS='KEEP' )
0200 C
0201 C Finished return
0202 C
0203      RETURN
0204 C
0205      END

```

COMMON2 .BLK

```

C
C   Common Block for Bumper Ver 4.0 5/25/87
C
C   ielm = max number of elements
C   ith  = max number of threats
C
C       PARAMETER (IELM=9000, ITH=400)
C
C       INTEGER*2 IT, ITYPE, NB, NC, NEL, NELM, NT, NV, EXPOSED(ITH),
1          POINT(IELM, ITH)
C
C       INTEGER*4 NR, ID(2, IELM), RANGE(2, 50)
C
C       REAL*4 BINC, CBETA, DIAM, VINC, AREA(IELM), GEOMETRY(IELM, ITH),
1          RESPONSE(70, 90, 5), THREAT(4, ITH)
C
C       REAL*8 ALT, ETIME, FLX
C
C       COMMON ALT, BINC, CBETA, DIAM, ETIME, FLX, IT, ITYPE, NB, NC, NEL, NELM,
1          NR, NT, NV, VR, VINC, AREA, EXPOSED, GEOMETRY, ID, POINT, RANGE,
2          RESPONSE, THREAT

```

This page left intentionally blank.

D180-30550-4

APPENDIX E

Source code for CONTOUR Module

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C C
0004 C CONTOUR VER 2.0 6/5/87 C
0005 C C
0006 C BOEING AEROSPACE CO. C
0007 C C
0008 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0009 C
0010 C
0011 C CONTOUR develops data for producing design contour plots for
0012 C use in designing spacecraft subject to meteoroid debris impact.
0013 C It is a modification to the BUMPER code. The modification involved
0014 C making the RESPONSE code a subroutine and expanding the user input.
0015 C The code ddevelops the data for a specific range of element ids
0016 C and a single property id.
0017 C
0018 C The code was developed under NASA contract ' Integrated Wall Design
0019 C Guide and Penetration Control Plan', by M.A.Wright and A.R.Coronado.
0020 C
0021 C
0022 C
0023 C
0024 C Common Block Variable list
0025 C
0026 C Scalars
0027 C
0028 C alt = operating altitude , km
0029 C binc = impact angle (beta) increment , deg
0030 C cbeta = cosine of beta
0031 C diam = critical diameter , cm
0032 C etime = spacecraft exposure time , years
0033 C flx = number of impacts per projected area per year of diameter D
0034 C or larger
0035 C itype = analysis type , 1- debris, 2-meteoroids
0036 C nb = number of angles in the response array
0037 C nel = current element number
0038 C nelm = total number of elements
0039 C nr = number of element ranges to sum over
0040 C nt = number of threat cases
0041 C nv = number of velocities in the response array
0042 C vr = impact (relative) velocity , km/sec
0043 C vinc = impact (relative) velocity increment , km/sec
0044 C
0045 C Arrays
0046 C
0047 C area = array containing the value of the surface area for each
0048 C element, sq-meters
0049 C exposed = list of the number of exposed elements for each threat
0050 C angle.
0051 C geometry = array containing the values of the cosine of the impact
0052 C angle for each exposed element for each threat angle.
0053 C id = array containing the values of the element and property id
0054 C for each element
0055 C 1- id
0056 C 2- pid
0057 C point = array of the element numbers corresponding to the elements
0058 C in the geometry array.
0059 C range = array containing the starting and ending element id for
0060 C each range to sum over
0061 C 1-starting id
0062 C 2- ending id
0063 C rtable = array containing the values of the critical diameter as

```

D180-30550-4

```

0064 C          a function of impact angle and velocity.
0065 C          (vr,beta,pid)
0066 C      threat = array containg threat information
0067 C          1-theta angle, rad
0068 C          2-phi angle, rad
0069 C          3-vr, km/sec
0070 C          4-prob
0071 C      history = list containing the  running sum of the fa term for each
0072 C          element
0073 C
0074 C
0075 C Main Program Variable List
0076 C
0077 C Scalers
0078 C
0079 C      answer = user input
0080 C      ctype = configuration type
0081 C          1- single plate
0082 C          2- double plate
0083 C      fa = product of the threat probability, flux and cosine(beta),
0084 C          impacts/yr/sq-meter
0085 C      kc = counter
0086 C      mli = logical variable used to determine if multi-layer
0087 C          insulation is to be included
0088 C      nst = number of shield thickness's
0089 C      nwt = number of vessel wall thickness's
0090 C      pfunc = penetration function
0091 C          1- original
0092 C          2- pen4
0093 C          3- regression
0094 C      pid = property id
0095 C      pnp = probabiltiy of no penetration, %
0096 C      prob = probabiltiy of threat case i occuring
0097 C      rat = ratio of the shiled thickness to total thickness
0098 C      shthk = shield thickness , in
0099 C      stand = shield stand-off distance, in
0100 C      sum = sum of the history array for the specific range
0101 C      tlinc = shield thickness increment , in
0102 C      tlmax = maximum shield thickness , in
0103 C      tlmin = minimum shield thickness , in
0104 C      t2inc = shield thickness increment , in
0105 C      t2max = maximum shield thickness , in
0106 C      t2min = minimum shield thickness , in
0107 C      vwthk = vessel wall thickness, in
0108 C
0109 C
0110 C Arrays
0111 C
0112 C      bhard = array containing the brinnel hardnes values for the shield
0113 C          vessell wall and projectile
0114 C      C = array containing the speed of sound values for the shield
0115 C          vessel wall and projectile, ft/sec
0116 C      dens = array containing the densisty of the shield vessel wall
0117 C          and projectile, lbs/in**3
0118 C      fsu = array containing the allowable shear stress for the shield
0119 C          vessel wall and projectile, psi
0120 C      ftu = array containing the allowable tensile ultimate stress for
0121 C          the shield vessel wall and projectile, psi
0122 C      fy = array containing the allowable yield stress for the shield
0123 C          vessel wall and projectile, psi
0124 C      wilkc = array containg Wilikinson's constant for the shield vessel
0125 C          wall and projectile
0126 C

```

D180-30550-4

```

0127 C
0128 INCLUDE 'COMMON3.BLK'
0151 C
0152 CHARACTER*80 ANSWER
0153 C
0154 INTEGER*2 CTYPE,PID,PFUNC
0155 C
0156 LOGICAL MLI
0157 C
0158 REAL*4 STAND,T1INC,T1MAX,T1MIN,T2INC,T2MAX,T2MIN,BHARD(3),C(3),
0159 1 DENS(3),FSU(3),FTU(3),FY(3),SHPV(3),WILKC(3)
0160 C
0161 REAL*8 FA,PROB,PNP,SUM,HISTORY(IEL)
0162 C
0163 C Write header to screen,
0164 C
0165 CALL HEADER ( ITYPE )
0166 C
0167 C Read in screen inputs
0168 C
0169 CALL INPUT ( CTYPE,MLI,PID,PFUNC,STAND,T1INC,T1MAX,T1MIN,T2INC,
0170 1 T2MAX,T2MIN,BHARD,C,DENS,FSU,FTU,FY,SHPV,WILKC )
0171 C
0172 C Read in the GEOMETRY output file
0173 C
0174 CALL GEOREAD
0175 C
0176 C Set the number of shield and vessel wall thickness cases
0177 C
0178 NST = NINT((T1MAX-T1MIN)/T1INC)+1
0179 NWT = NINT((T2MAX-T2MIN)/T2INC)+1
0180 C
0181 C Initialize counter to 1
0182 C
0183 KC = 1
0184 C
0185 C Loop through the various wall configurations
0186 C
0187 DO 1000 I=1,NWT
0188 C
0189 C Set vessel wall thickness
0190 C
0191 VWTHK=T2MIN+(I-1)*T2INC
0192 C
0193 DO 900 J=1,NST
0194 C
0195 C Set the shield thickness
0196 C
0197 SHTHK=T1MIN+(J-1)*T1INC
0198 C
0199 C Check that configuration is within the limits
0200 C
0201 IF ( CTYPE.NE.1 ) THEN
0202 RAT=SHTHK/(SHTHK+VWTHK)
0203 IF ( RAT.GT.0.50 .OR. RAT.LT.0.10 ) GO TO 900
0204 END IF
0205 C
0206 C Build the response array
0207 C
0208 CALL RESPONSE (BINC,CTYPE,ITYPE,MLI,NB,NV,PID,PFUNC,SHTHK,
0209 1 STAND,VWTHK,VINC,BHARD,C,DENS,FSU,FTU,FY,
0210 2 RTABLE,SHPV,WILKC )
0211 C

```

D180-30550-4

```

0212 C Initialize History to 0.0
0213 C
0214 C DO 50 K=1,NELM
0215 C HISTORY(K)=0.0D0
0216 50 CONTINUE
0217 C
0218 C Initialize Sum to 0.0
0219 C
0220 C SUM =0.0D0
0221 C
0222 C Determine the penetrating flux for each element, for each threat
0223 C angle
0224 C
0225 C DO 200 K=1,NT
0226 C
0227 C Set the threat index and get the impact velocity and the threat
0228 C probability from the threat array
0229 C
0230 C VR=THREAT(3,K)
0231 C PROB=THREAT(4,K)
0232 C
0233 C Evaluate each exposed element
0234 C
0235 C DO 100 L=1,EXPOSED(K)
0236 C
0237 C Set the element number
0238 C
0239 C NEL=POINT(L,K)
0240 C
0241 C IF ( RANGE(1).GT.ID(1,NEL) ) GO TO 100
0242 C IF ( RANGE(2).LT.ID(1,NEL) ) GO TO 100
0243 C
0244 C Check that property id specified matches elements property id
0245 C if not skip over element.
0246 C
0247 C IF ( PID.NE.ID(2,NEL) ) THEN
0248 C GO TO 100
0249 C END IF
0250 C
0251 C Get the cosine of the impact angle from the Geometry array.
0252 C
0253 C CBETA=GEOMETRY(L,K)
0254 C
0255 C Determine the diameter of the sphere that just penetrates the wall
0256 C
0257 C CALL CRITDIA
0258 C
0259 C Calculate the flux of the critical diameter using the appropriate
0260 C flux equation based on the analysis type.
0261 C
0262 C CALL FLUX
0263 C
0264 C Multiply the flux by the probability of the threat angle and the
0265 C cosine of the impact angle, this determines the number of
0266 C penetrations per year per element surface area for the current
0267 C element and the current threat angle.
0268 C
0269 C FA=FLX*PROB*CBETA
0270 C
0271 C Store the running sum of the FA term for each element in the History
0272 C array. This value represents the total number of penetrations for
0273 C a given element per year per element surface area.
0274 C

```

D180-30550-4

```

0275          HISTORY(NEL)=HISTORY(NEL)+FA
0276      C
0277      C  Next element
0278      C
0279      100          CONTINUE
0280      C
0281      C  Next threat
0282      C
0283      200          CONTINUE
0284      C
0285      C  Multiply each term in the HISTORY array by the appropriate element
0286      C  surface area.
0287      C
0288          DO 250 K=1,NELM
0289              HISTORY(K)=HISTORY(K)*AREA(K)
0290      250          CONTINUE
0291      C
0292      C  Sum up the HISTORY array by range
0293      C
0294          DO 310 K=1,NELM
0295              IF ( RANGE(1).GT.ID(1,K) ) GO TO 310
0296              IF ( RANGE(2).LT.ID(1,K) ) GO TO 310
0297              IF ( PID.NE.ID(2,K) ) GO TO 310
0298              SUM=SUM+HISTORY(K)
0299      310          CONTINUE
0300      C
0301      C  Calculate the probability of no penetration for each range using
0302      C  a Poission model
0303      C
0304          PNP=(DEXP(-SUM*ETIME))*100.D0
0305      C
0306      C  Write the probability value to the screen and the summary file.
0307      C
0308          WRITE ( 6,390 ) KC,SHTHK,STAND,VWTHK,PNP
0309          WRITE ( 10,390 ) KC,SHTHK,STAND,VWTHK,PNP
0310      390          FORMAT ( /1X,I4,3F7.4,F12.5 )
0311      C
0312          KC=KC+1
0313      C
0314      C  Next shield thickness
0315      C
0316      900          CONTINUE
0317      C
0318      C  Next vessel wall thickness
0319      C
0320      1000 CONTINUE
0321      C
0322      C  Close summary file
0323      C
0324          CLOSE ( UNIT=10,STATUS='KEEP' )
0325      C
0326      C  Finished
0327      C
0328          END

```

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE HEADER ( ITYPE )
0005 C
0006 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007 C
0008 C
0009 C Header writes out the program header and reads in the summary file
0010 C name and analysis type ( meteoroid or debris ).
0011 C
0012 C Variable list
0013 C
0014 C answer = user input
0015 C
0016 CHARACTER*80 ANSWER
0017 C
0018 C Write Header to screen
0019 C
0020 WRITE ( 6,10 )
0021 10 FORMAT ( /1X,'*****',//,5X,'CONTOUR ',
0022 1 'Version 2.0',//,5X,'last update 6/5/87',//1X,
0023 2 '*****' )
0024 C
0025 C Read in summary output filename, set default to contour.sum
0026 C
0027 20 WRITE ( 6,30 )
0028 30 FORMAT ( /1X,'SUMMARY OUTPUT FILENAME (CR=CONTOUR.SUM) : ', $)
0029 READ ( 5,'(A)' )ANSWER
0030 C
0031 IF ( ANSWER(1:1).EQ.' ' ) ANSWER='CONTOUR.SUM'
0032 C
0033 C Open sfile
0034 C
0035 OPEN ( UNIT=10,FILE=ANSWER,STATUS='UNKNOWN',ERR=40 )
0036 C
0037 REWIND 10
0038 C
0039 GO TO 60
0040 C
0041 C Error control
0042 C
0043 40 WRITE ( 6,50 )
0044 50 FORMAT (/1X,'UNABLE TO OPEN FILE ' )
0045 GO TO 20
0046 C
0047 C Determine analysis type, set default to 1 (debris)
0048 C
0049 60 WRITE ( 6,70 )
0050 70 FORMAT (/1X,'ANALYSIS TYPE ?',/,2X,'1-DEBRIS <CR> ',/,2X,
0051 1 '2-METEORIDS',//,1X,'ANSWER 1 OR 2 : ', $)
0052 READ ( 5,'(A)' ) ANSWER
0053 C
0054 IF ( ANSWER(1:1).EQ.' ' ) THEN
0055 ITYPE=1
0056 ELSE
0057 READ ( ANSWER(1:4),'(BN,I4)',ERR=60 )ITYPE
0058 END IF
0059 C
0060 C Check that input was correct
0061 C
0062 IF ( ITYPE.EQ.1 .OR. ITYPE.EQ.2 ) THEN
0063 CONTINUE

```

```
0064      ELSE
0065          WRITE ( 6,80 )
0066      80    FORMAT ( /1X,'INCORRECT INPUT' )
0067          GO TO 60
0068      END IF
0069  C
0070  C  Write analysis type and etime to summary file
0071  C
0072      WRITE ( 10,10 )
0073  C
0074      IF ( ITYPE.EQ.1 ) THEN
0075          WRITE ( 10,90 )
0076      90    FORMAT ( /1X,'MAN-MADE ORBITAL DEBRIS ANALYSIS')
0077      ELSE
0078          WRITE ( 10,100 )
0079      100    FORMAT ( 1X,'METEOROID ANALYSIS' )
0080      END IF
0081  C
0082  C  Finished
0083  C
0084      RETURN
0085  C
0086      END
```

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE INPUT ( CTYPE,MLI,PID,PFUNC,STAND,T1INC,T1MAX,T1MIN,
0005 1 T2INC,T2MAX,T2MIN,BHARD,C,DENS,FSU,FTU,FY,
0006 2 SHPV,WILKC )
0007 C
0008 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0009 C
0010 C Input reads in the user defined inputs and writes them out to the
0011 C summary file. It is a modified version of the combined Resposne and
0012 C Bumper codes input subroutines.
0013 C
0014 C For variables in the common block see main listing
0015 C
0016 C Variable list
0017 C
0018 C ncases = potential numeber of wall configurations
0019 C metric = logical variable used to determine if input units are in
0020 C the metric system
0021 C lc = number of materials red in from the MAT.PRP file
0022 C answer = user input
0023 C luniits = lenght units in or cm
0024 C
0025 C Array list
0026 C
0027 C dns = temporary array containing the density values in the MAT.PRP
0028 C file, lbs/in**3
0029 C con = temporary array containing the Wilkinson's constant in the
0030 C MAT.PRP file, km/sec
0031 C hrd = temporary array containing Brinnel Hardness values in the
0032 C MAT.PRP file, unitless
0033 C shr = temporary array containing the Shear stress allowable values
0034 C in the MAT.PRP file , psi
0035 C shk = temporary array containg the shock projectile velocities in
0036 C in the MAT.PRP file
0037 C ssnd = temporary array containing the speed of sound values in the
0038 C MAT.PRP file, ft/sec
0039 C ult = tempoary array containing the ultimate tensile stress
0040 C allowables in the MAT.PRP file
0041 C yld = temporary array containing yield stress allowables in the
0042 C MAT.PRP file
0043 C
0044 C
0045 C INCLUDE 'COMMON3.BLK'
0068 C
0069 C
0070 C INTEGER*2 CTYPE,PID,PFUNC
0071 C
0072 C LOGICAL MLI
0073 C
0074 C REAL*4 STAND,T1INC,T1MAX,T1MIN,T2INC,T2MAX,T2MIN,BHARD(3),C(3),
0075 1 DENS(3),FSU(3),FTU(3),FY(3),SHPV(3),WILKC(3)
0076 C
0077 C CHARACTER*2 LUNITS
0078 C CHARACTER*12 MNAM(10)
0079 C CHARACTER*20 ANSWER
0080 C
0081 C INTEGER*2 LC,MAT(3)
0082 C
0083 C LOGICAL METRIC
0084 C
0085 C REAL*4 DNS(10),CON(10),HRD(10),SHR(10),SHK(10),SSND(10),ULT(10),

```

D180-30550-4

```

0086      1      YLD(10)
0087      C
0088      C Initialize variables
0089      C
0090      LC=1
0091      MLI=.TRUE.
0092      METRIC=.TRUE.
0093      C
0094      C Determine the spacecraft exposure time , set default to 10 years
0095      C and write out to summary file
0096      C
0097      10 WRITE ( 6,20 )
0098      20 FORMAT (/1X,'SPACECRAFT EXPOSURE TIME (YEARS) <CR=10.0> : ', $)
0099      READ ( 5,'(A)' ) ANSWER
0100      C
0101      IF ( ANSWER(1:1).EQ.' ' ) THEN
0102          ETIME=10.0D0
0103      ELSE
0104          READ ( ANSWER(1:20),'(BN,D20.0)',ERR=10 ) ETIME
0105      END IF
0106      C
0107      WRITE ( 10,30 ) ETIME
0108      30 FORMAT ( /1X,'SPACECRAFT EXPOSURE TIME (YEARS) =',F7.2 )
0109      C
0110      C Read in operating altitude , set default to 500 km, and write out to
0111      C summary file
0112      C
0113      40 WRITE ( 6,50 )
0114      50 FORMAT ( /1X,'OPERATING ALTITUDE ( 400-500 km ) <CR=500> : ', $)
0115      READ ( 5,'(A)' ) ANSWER
0116      C
0117      IF ( ANSWER(1:1).EQ.' ' ) THEN
0118          ALT=500.0D0
0119      ELSE
0120          READ ( ANSWER(1:20),'(BN,D20.0)',ERR=40 ) ALT
0121      END IF
0122      C
0123      C Check that altitude is within range
0124      C
0125      IF ( ALT.LT.350.0 .OR. ALT.GT.550.0 ) THEN
0126          WRITE ( 6,60 )
0127      60 FORMAT ( 1X,'---ERROR--- Altitude outside of range' )
0128          GO TO 40
0129      END IF
0130      C
0131      WRITE ( 10,70 ) ALT
0132      70 FORMAT ( /1X,'OPERATING ALTITUDE (km) = ',F7.2 )
0133      C
0134      C Read in element range to sum over
0135      C
0136      WRITE ( 6,80 )
0137      80 FORMAT ( /1X,'THE PROBABILITY OF NO PENETRATION WILL BE ',
0138          1      'CALCULATED FOR A SPECIFIC RANGE',/1X,'OF ELEMENT IDS ',
0139          2      'INPUT THE STARTING AND ENDING ELEMENT ID FOR EACH ',
0140          3      'RANGE')
0141      C
0142      90 WRITE ( 6,100 )
0143      100 FORMAT ( /1X,'STARTING ELEMENT ID : ', $)
0144      READ ( 5,'(A)' ) ANSWER
0145      C
0146      READ ( ANSWER(1:20),'(BN,I8)',ERR=90 ) RANGE(1)
0147      C
0148      110 WRITE ( 6,120 )

```

```

0149      120 FORMAT ( 1X,'ENDING ELEMENT ID   : ', $)
0150      READ ( 5,'(A)' ) ANSWER
0151      C
0152      READ ( ANSWER(1:20),'(BN,I8)',ERR=110 ) RANGE(2)
0153      C
0154      C Check that ending id > starting id
0155      C
0156      IF ( RANGE(1).GT.RANGE(2) ) THEN
0157          WRITE ( 6,130 )
0158      130  FORMAT ( 1X,'---ERROR--- Staring ID greater then Ending ID')
0159          GO TO 90
0160      END IF
0161      C
0162      C Write range out to summary file
0163      C
0164          WRITE ( 10,140 ) RANGE(1),RANGE(2)
0165      140  FORMAT ( /1X,'STARTING ELEMENT ID =',I8,/1X,'ENDING ELEMENT ID ',
0166          1      ' =',I8 )
0167      C
0168      C Read in the material properties file.
0169      C
0170          OPEN (UNIT=2,FILE='MAT.PRP',STATUS='OLD')
0171      C
0172      150  READ ( 2,'(A,BN,8E12.5)',END=160 ) MNAM(LC),DNS(LC),YLD(LC),
0173          1      ULT(LC),SHR(LC),CON(LC),
0174          2      SSND(LC),SHK(LC),HRD(LC)
0175      C
0176          LC=LC+1
0177          GO TO 150
0178      C
0179      160  CLOSE ( UNIT=2,STATUS='KEEP' )
0180          LC=LC-1
0181      C
0182      C Determine the type of units for input.
0183      C
0184          WRITE ( 6,170 )
0185      170  FORMAT (/1X,'INPUT IN METRIC OR ENGLISH UNITS <CR>=METRIC : ', $)
0186          READ ( 5,'(A)' ) ANSWER
0187      C
0188          IF ( ANSWER(1:1).EQ.'E' ) METRIC=.FALSE.
0189      C
0190      C Set the units
0191      C
0192      180  IF ( METRIC ) THEN
0193          LUNITS='CM'
0194      ELSE
0195          LUNITS='IN'
0196      END IF
0197      C
0198      C Read in property id
0199      C
0200      185  WRITE ( 6,190 )
0201      190  FORMAT ( /1X,'PROPERTY ID = <CR=1> : ', $)
0202          READ ( 5,'(A)' ) ANSWER
0203      C
0204          IF ( ANSWER(1:1).EQ.' ' ) THEN
0205              PID=1
0206          ELSE
0207              READ ( ANSWER(1:20),'(BN,I4)',ERR=185 ) PID
0208          END IF
0209      C
0210          IF ( PID.GT.10 .OR. PID.LT.1 ) THEN
0211              WRITE ( 6,'(/1X,'INCORRECT INPUT' )' )

```

```

0212          GO TO 185
0213      END IF
0214      C
0215          WRITE ( 10, '( /1X, 'PROPERTY ID = ', I4 ) ' ) PID
0216      C
0217      C Determine configuration type and write out to summary file
0218      C
0219          200 WRITE ( 6, 210 )
0220          210 FORMAT ( /1X, 'CONFIGURATION TYPE', /, 5X, '1- SINGLE PLATE ', /,
0221              1      5X, '2- DOUBLE PLATE <CR>', /, 1X, 'ANSWER (1 or 2) : ', $)
0222          READ ( 5, '(A)' ) ANSWER
0223      C
0224          IF ( ANSWER(1:1).EQ.' ' ) THEN
0225              CTYPE=2
0226          ELSE
0227              READ ( ANSWER(1:4), '(BN, I4)', ERR=200 ) CTYPE
0228          END IF
0229      C
0230      C Check that the input was correct
0231      C
0232          IF ( CTYPE.LT.1 .OR. CTYPE.GT.2 ) THEN
0233              WRITE ( 6, 220 )
0234          220      FORMAT( /1X, 'INCORRECT INPUT' )
0235              GO TO 200
0236          END IF
0237      C
0238          IF ( CTYPE.EQ.1 ) THEN
0239              WRITE ( 10, 230 )
0240          230      FORMAT ( /1X, 'SINGLE PLATE ' )
0241          ELSE
0242              WRITE ( 10, 240 )
0243          240      FORMAT ( /1X, 'DOUBLE PLATE ANALYSIS' )
0244          END IF
0245      C
0246      C For single plate configuration skip down to the vessel wall material
0247      C
0248          IF ( CTYPE.EQ.1 ) GO TO 430
0249      C
0250      C Determine which double wall penetration function to use and write
0251      C out to summary file
0252      C
0253          250 WRITE ( 6, 260 )
0254          260 FORMAT ( /1X, 'PENETRATION FUNCTION ', /, 5X, '1-ORIGINAL <CR>', /,
0255              1      5X, '2-PEN4', /, 5X, '3-REGRESSION', /, 1X, 'ANSWER (1-3) : ',
0256              2      $)
0257      C
0258          READ ( 5, '(A)' ) ANSWER
0259          IF ( ANSWER(1:1).EQ.' ' ) THEN
0260              PFUNC=1
0261          ELSE
0262              READ ( ANSWER(1:20), '(BN, I4)', ERR=250 ) PFUNC
0263          END IF
0264      C
0265      C Check Input
0266      C
0267          IF ( PFUNC.LT.1 .OR. PFUNC.GT.3 ) GO TO 250
0268      C
0269          IF ( PFUNC.EQ.1 ) THEN
0270              WRITE ( 10, 270 )
0271          270      FORMAT ( /1X, 'ORIGINAL PENETRATION FUNCTION' )
0272          ELSE IF ( PFUNC.EQ.2 ) THEN
0273              WRITE ( 10, 280 )
0274          280      FORMAT ( /1X, 'PEN4 PENETRATION FUNCTION' )

```

```

0275         ELSE IF ( PFUNC.EQ.3 ) THEN
0276             WRITE ( 10,290 )
0277     290     FORMAT ( /1X,'REGRESSION PENETRATION FUNCTION' )
0278         END IF
0279     C
0280     C Determine the shield material.
0281     C
0282     300 WRITE ( 6,310 )
0283     310 FORMAT (/1X,'SHIELD MATERIAL ')
0284     C
0285     C Write out the material list.
0286     C
0287         DO 320 I=1,LC
0288             WRITE ( 6,315 ) I,MNAM(I)
0289     315     FORMAT ( 3X,I2,'- ',A )
0290     320 CONTINUE
0291     C
0292     C Set the material default number equal to one.
0293     C
0294     330 WRITE ( 6,340 )
0295     340 FORMAT (1X,'SELECT MATERIAL NUMBER <CR>=1 : ', $)
0296         READ ( 5,'(A)' ) ANSWER
0297     C
0298         IF ( ANSWER(1:1).EQ.' ' ) THEN
0299             MAT(1)=1
0300         ELSE
0301             READ ( ANSWER(1:4),'(BN,I4)',ERR=330 ) MAT(1)
0302         END IF
0303     C
0304     C Check that the value read in is contained in the list.
0305     C
0306         IF ( MAT(1) .LT.1 .OR. MAT(1).GT. LC ) GO TO 330
0307     C
0308     C Write shield material out to summary file
0309     C
0310         WRITE ( 10,350 ) MNAM(MAT(1))
0311     350 FORMAT ( /1X,'SHILED MATERIAL = ',A )
0312     C
0313     C Determine the shield minimum, maximum, and increment thickness.
0314     C
0315     360 WRITE ( 6,370 ) LUNITS
0316     370 FORMAT ( /1X,'MINIMUM SHIELD THICKNESS (' ,A,') = : ', $)
0317         READ ( 5,*,ERR=360 ) T1MIN
0318     C
0319     380 WRITE ( 6,390 ) LUNITS
0320     390 FORMAT ( /1X,'MAXIMUM SHIELD THICKNESS (' ,A,') = : ', $)
0321         READ ( 5,*,ERR=380 ) T1MAX
0322     C
0323     400 WRITE ( 6,410 ) LUNITS
0324     410 FORMAT ( /1X,'INCREMENT SHIELD THICKNESS (' ,A,') = : ', $)
0325         READ ( 5,*,ERR=400 ) T1INC
0326     C
0327     C Write out values to summary file
0328     C
0329         WRITE ( 10,420 ) LUNITS,T1MIN,T1MAX,T1INC
0330     420 FORMAT ( /1X,'MINIMUM MAXIMUM AND INCREMENT SHIELD THICKNESS (' ,
0331     1         A,') = ',/1X,3F7.4 )
0332     C
0333     C Determine the vessel wall material.
0334     C
0335     430 WRITE ( 6,440 )
0336     440 FORMAT (/1X,'VESSEL WALL MATERIAL ' )
0337     C

```

```

0338      DO 450 I=1,LC
0339      WRITE ( 6,315 ) I,MNAM(I)
0340      450 CONTINUE
0341      C
0342      460 WRITE ( 6 ,340 )
0343      READ ( 5,'(A)' ) ANSWER
0344      C
0345      IF ( ANSWER(1:1) .EQ. ' ' ) THEN
0346      MAT(2)=1
0347      ELSE
0348      READ ( ANSWER (1:4),'(BN,I4)',ERR=460 ) MAT(2)
0349      END IF
0350      C
0351      IF ( MAT(2).LT.1 .OR. MAT(2).GT.LC ) GO TO 460
0352      C
0353      C Write vessel wall material out to summary file
0354      C
0355      WRITE ( 10,470 ) MNAM(MAT(2))
0356      470 FORMAT ( /1X,'VESSEL WALL MATERIAL = ',A )
0357      C
0358      C Determine the minimum , maximum, and increment vessel wall thickness
0359      C
0360      480 WRITE ( 6,490 ) LUNITS
0361      490 FORMAT (/1X,'MINIMUM VESSEL WALL THICKNESS (' ,A,') = : ', $)
0362      READ ( 5,*,ERR=480 ) T2MIN
0363      C
0364      500 WRITE ( 6,510 ) LUNITS
0365      510 FORMAT (/1X,'MAXIMUM VESSEL WALL THICKNESS (' ,A,') = : ', $)
0366      READ ( 5,*,ERR=500 ) T2MAX
0367      C
0368      520 WRITE ( 6,530 ) LUNITS
0369      530 FORMAT (/1X,'INCREMENT VESSEL WALL THICKNESS (' ,A,') = : ', $)
0370      READ ( 5,*,ERR=520 ) T2INC
0371      C
0372      C Check if number of potential cases is ok
0373      C
0374      IF ( T1INC.EQ.0.0 ) T1INC=1.0
0375      NCASES=((T2MAX-T2MIN)/T2INC+1)*((T1MAX-T1MIN)/T1INC+1)
0376      C
0377      WRITE ( 6,540 ) NCASES
0378      540 FORMAT (/1X,'THE NUMBER OF POTENTIAL CASES IS ',I6,/1X,'DO YOU ',
0379      1 'WISH TO CONTINUE ? <CR=YES> : ', $)
0380      READ ( 5,'(A)' ) ANSWER
0381      C
0382      IF ( ANSWER(1:1).EQ.'N' ) GO TO 300
0383      C
0384      C Write values out to the summary file
0385      C
0386      WRITE ( 10,550 ) LUNITS,T2MIN,T2MAX,T2INC
0387      550 FORMAT ( /1X,'MINIMUM MAXIMUM AND INCREMENT VESSEL WALL ',
0388      1 'THICKNESS (' ,A,') = ',/1X,3F7.4 )
0389      C
0390      C For single plate configuration skip stand-off and mli
0391      C
0392      IF ( CTYPE.EQ.1 ) GO TO 610
0393      C
0394      C Determine the shield stand-off distance and write out to the summary
0395      C file.
0396      C
0397      560 WRITE ( 6,570 ) LUNITS
0398      570 FORMAT ( /1X,'SHIELD STAND-OFF (' ,A,') = : ', $)
0399      READ ( 5,*,ERR=560 ) STAND
0400      C

```

D180-30550-4

```

0401      WRITE ( 10,580 ) LUNITS,STAND
0402 580 FORMAT ( /1X,'SHIELD STAND-OFF (' ,A,' ) = ' ,F12.5 )
0403 C
0404 C Determine if MLI is to be included, but not for the pen4 penetration
0405 C function
0406 C
0407      IF ( PFUNC.EQ.2 ) GO TO 610
0408 C
0409      WRITE ( 6,590 )
0410 590 FORMAT (/1X,'INCLUDE 30 LAYERS OF MLI BETWEEN PLATES <CR>=Y : ',
0411 1      $)
0412      READ ( 5,'(A)' )ANSWER
0413      IF ( ANSWER(1:1).EQ.' ' .OR. ANSWER(1:1).EQ.'Y' )MLI=.TRUE.
0414 C
0415      IF ( MLI ) THEN
0416          WRITE ( 10,600 )
0417 600 FORMAT ( /1X,'30 LAYERS OF MULTI-LAYER INSULATION INCLUDED' )
0418      END IF
0419 C
0420 C Set the projectile material property based on analysis type
0421 C For debris 100-0 al, for meteoroids only density is important use
0422 C use 0.50 g/cc .
0423 C
0424 610 IF ( ITYPE.EQ.1 ) THEN
0425      BHARD(3)=23.0
0426      C(3)=16550.0
0427      DENS(3)=0.098
0428      FSU(3)=5000.0
0429      FTU(3)=13000.0
0430      FY(3)=9000.0
0431      SHPV(3)=1.345
0432      WILKC(3)=.126
0433      ELSE
0434      BHARD(3)=0.0
0435      C(3)=0.0
0436      DENS(3)=0.50/27.705
0437      FSU(3)=0.0
0438      FTU(3)=0.0
0439      FY(3)=0.0
0440      SHPV(3)=0.0
0441      WILKC(3)=0.0
0442      END IF
0443
0444 C
0445 C Build the material properties arrays.
0446 C
0447      DO 650 I=1,2
0448          BHARD(I)=HRD(MAT(I))
0449          C(I)=SSND(MAT(I))
0450          DENS(I)=DNS(MAT(I))
0451          FSU(I)=SHR(MAT(I))
0452          FTU(I)=ULT(MAT(I))
0453          FY(I)=YLD(MAT(I))
0454          SHPV(I)=SHK(MAT(I))
0455          WILKC(I)=CON(MAT(I))
0456 650 CONTINUE
0457 C
0458 C If the variables were read in in metric units, convert to english.
0459 C
0460      IF ( METRIC ) THEN
0461          T1MIN=T1MIN/2.54
0462          T1MAX=T1MAX/2.54
0463          T1INC=T1INC/2.54

```

D180-30550-4

```
0464      T2MIN=T2MIN/2.54
0465      T2MAX=T2MAX/2.54
0466      T2INC=T2INC/2.54
0467      STAND=STAND/2.54
0468      END IF
0469      C
0470      RETURN
0471      C
0472      END
```

D180-30550-4

```

0001      C
0002      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003      C
0004          SUBROUTINE GEOREAD
0005      C
0006      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007      C
0008      C
0009      C   Georead reads in the output file from the GEOMETRY code. This file
0010      C   contains the global threat and element data as well as the list of
0011      C   exposed elements and their impact angles for each threat case.
0012      C
0013      C
0014      C
0015      C   note: for variables contained in the common block refer to main
0016      C   listing for definition
0017      C
0018      C
0019      C   Variable List
0020      C
0021      C       answer = character string representing user input
0022      C       gfile = geometry output filename
0023      C       itf = analysis type contained in the
0024      C
0025      C
0026          INCLUDE 'COMMON3.BLK'
0049      C
0050          CHARACTER*80 ANSWER,GFILE
0051      C
0052          INTEGER*2 ITF
0053      C
0054      C   Read in the GEOMETRY output filename, set the default to geom.dat
0055      C
0056          10 WRITE ( 6,20 )
0057          20 FORMAT(/1X,'GEOMETRY OUTPUT FILENAME ? <CR=STATION.GEM> >',$)
0058          READ ( 5,30 ) GFILE
0059          30 FORMAT (A)
0060      C
0061          IF ( GFILE(1:1).EQ.' ' ) GFILE='STATION.GEM'
0062      C
0063      C   Open the file
0064      C
0065          OPEN (UNIT=2,FILE=GFILE,STATUS='OLD',FORM='UNFORMATTED',ERR=40 )
0066      C
0067          GO TO 60
0068      C
0069      C   Error control
0070      C
0071          40 WRITE ( 6,50 )
0072          50 FORMAT ( /1X,'UNABLE TO OPEN FILE ' )
0073          GO TO 10
0074      C
0075      C   Read in the analysis type,the number of threat cases, and the
0076      C   number of elements
0077      C
0078          60 READ (2) ITF,NT,NELM
0079      C
0080      C   Check that the analysis type in the file matches the type input
0081      C   by user.
0082      C
0083          IF ( ITF.NE.ITYPE ) THEN
0084              IF ( ITYPE.EQ.1 ) THEN
0085                  WRITE ( 6,70 )

```

D180-30550-4

```

0086      70      FORMAT ( /1X,'DEBRIS ANALYSIS SPECIFIED BUT FILE IS FOR ',
0087      1          'METEOROIDS')
0088      ELSE
0089      WRITE ( 6,80 )
0090      80      FORMAT ( /1X,'METEOROID ANALYSIS SPECIFIED BUT FILE IS ',
0091      1          'FOR DEBRIS' )
0092      END IF
0093      C
0094      WRITE ( 6,90 )
0095      90      FORMAT ( /1X,'DO YOU WISH TO CONTINUE (CR=NO) >','$)
0096      READ ( 5,30 ) ANSWER
0097      C
0098      IF ( ANSWER(1:1).EQ.'Y' ) THEN
0099      GO TO 10
0100      ELSE
0101      STOP
0102      END IF
0103      C
0104      END IF
0105      C
0106      C Check that the number of threats and the number of elements are less
0107      C then the maximum allowed
0108      C
0109      IF ( NT.GT.ITH ) THEN
0110      WRITE ( 6,100 )
0111      100      FORMAT ( /1X,'NUMBER OF THREATS IS GREATER THEN MAX ALLOWED')
0112      WRITE ( 6,105 )
0113      105      FORMAT ( 1X,'ARRAY SIZE MUST BE INCREASED & CODE RECOMPILED')
0114      STOP
0115      END IF
0116      C
0117      IF ( NELM.GT.IEL ) THEN
0118      WRITE ( 6,110 )
0119      110      FORMAT ( /1X,'NUMBER OF ELEMENTS IS GREATER THEN MAX ALLOWED')
0120      WRITE ( 6,105 )
0121      STOP
0122      END IF
0123      C
0124      C Initialize the arrays to 0.0
0125      C
0126      DO 150 I=1,NT
0127      THREAT(3,I)=0.0
0128      THREAT(4,I)=0.0
0129      EXPOSED(I)=0
0130      DO 140 J=1,NELM
0131      GEOMETRY(J,I)=0.0
0132      ID(1,J)=0
0133      ID(2,J)=0
0134      POINT(J,I)=0
0135      140      CONTINUE
0136      150      CONTINUE
0137      C
0138      C Read in the Threat data
0139      C
0140      DO 175 I=1,NT
0141      READ (2) (THREAT(J,I),J=1,4)
0142      175      CONTINUE
0143      C
0144      C Read in the element id, and property id storing them in the ID
0145      C array.
0146      C
0147      DO 200 I=1,NELM
0148      READ (2) (ID(J,I),J=1,2)

```

D180-30550-4

```

0149      200 CONTINUE
0150      C
0151      C   Read in the element's surface area storing it in the AREA array.
0152      C
0153          DO 250 I=1,NELM
0154              READ (2) AREA(I)
0155      250 CONTINUE
0156      C
0157      C   Read in the geometry data for the exposed elements
0158      C
0159          DO 500 I=1,NT
0160      C
0161      C   Read in the threat case and the number of exposed elements
0162      C
0163          READ (2) IT,EXPOSED(I)
0164      C
0165      C   Loop thru the exposed elements
0166      C
0167          DO 400 J=1,EXPOSED(I)
0168      C
0169      C   Read in the element number (storing in the POINT array), and the
0170      C   cosine of the impact angle (storing in the GEOMETRY array).
0171      C
0172          READ (2) POINT(J,I),GEOMETRY(J,I)
0173      C
0174      400      CONTINUE
0175      C
0176      500 CONTINUE
0177      C
0178      C   Write gfile to summary file
0179      C
0180          WRITE ( 10,600 )GFILE
0181      600 FORMAT ( 1X,'GEOMETRY OUTPUT FILE = ',A )
0182      C
0183      C   Close file
0184      C
0185          CLOSE ( UNIT=2,STATUS='KEEP' )
0186      C
0187          RETURN
0188      C
0189          END

```

```

0001      C                               D180-30550-4
0002      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003      C
0004      SUBROUTINE RESPONSE (AINCR,CTYPE,ITYPE,MLI,NANG,NVEL,PID,PFUNC,
0005      1                      SHTHK,STAND,VWTHK,VINCR,BHARD,C,DENS,FSU,
0006      2                      FTU,FY,RTABLE,SHPV,WILKC )
0007      C
0008      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0009      C
0010      C
0011      C  Variable list
0012      C
0013      C    aincr = impact angle increment,deg
0014      C    amax = maximum impact angle,deg
0015      C    amin = minimum impact angle,deg
0016      C    ang = impact angle,deg
0017      C    angr = impact angle,radians
0018      C    answer = character string representing user input
0019      C    ctype = configuration type
0020      C                      1- single plate
0021      C                      2 - double plate
0022      C    dia = projectile diameter,in
0023      C    diam = " " ,cm
0024      C    ic = case counter
0025      C    initial = logical variable used to determine if current call to
0026      C                diameter is the initial one for the current angle
0027      C    itype = analysis type,1=space debris
0028      C                2=meteoroids
0029      C    mli = logical variable used to determine if 30 layers of mli is
0030      C            included
0031      C    nang = number of angles to be considered
0032      C    nvel = number of velocities to be considered
0033      C    pfunc = penetration function
0034      C            1-original
0035      C            2-pen4
0036      C            3-regression
0037      C    shthk = shield thickness,in
0038      C    stand = shield stand-off,in
0039      C    vele = impact velocity,ft/sec
0040      C    velm = " " ,km/sec
0041      C    vincr = velocity increment,km/sec
0042      C    vmax = maximum velocity,km/sec
0043      C    vmin = minimum velocity,km/sec
0044      C
0045      C
0046      C  Array list
0047      C
0048      C    bhard = array containing the Brinnel hardness values for the
0049      C            current configuration
0050      C    c = array containing the speed of sound values for the current
0051      C            configuration, ft/sec
0052      C    dens = array containing the density values for the current
0053      C            configuration,lbs/in**3
0054      C    fsu = array containing the shear allowable stress values for the
0055      C            current configuration,psi
0056      C    ftu = array containing the ultimate tensile stress values for the
0057      C            current configuration,psi
0058      C    fy = array containing the yield stress values for the current
0059      C            configuration
0060      C    rtable = array containing the critical diameters for each case,angle
0061      C            and velocity
0062      C    shpv =array containing the shock projectile velocities for the
0063      C            current configuration

```

D180-30550-4

```

0064 C wilkc = array containing the values for Wilkinson's constant for
0065 C the current configuration, km/sec
0066 C
0067 C
0068 C
0069 C
0070 CHARACTER*20 ANSWER
0071 C
0072 DIMENSION RTABLE(70,50,10)
0073 C
0074 DIMENSION BHARD(3),C(3),DENS(3),FSU(3),FTU(3),FY(3),SHPV(3),
0075 1 WILKC(3)
0076 C
0077 INTEGER*2 CTYPE,IC,ITYPE,NANG,NVEL,PID,PFUNC
0078 C
0079 LOGICAL INITIAL,METRIC,MLI
0080 C
0081 C Set the angle and velocity limits and increments
0082 C
0083 C Amin must always =0
0084 C
0085 AMIN=0.0
0086 C
0087 AMAX=90.0
0088 AINCR=5.0
0089 C
0090 C Determine the number of velocity and angle iterations
0091 C
0092 NANG = (AMAX-AMIN)/AINCR + 1
0093 C
0094 C Set velocity values based on analysis type
0095 C
0096 C Vmin must always = 0
0097 C
0098 VMIN=0.0
0099 C
0100 IF ( ITYPE.EQ.1 ) THEN
0101 VMAX=17.0
0102 VINCR=0.25
0103 ELSE
0104 VMAX=70.0
0105 VINCR=1.0
0106 END IF
0107 C
0108 NVEL=(VMAX-VMIN)/VINCR
0109 C For the current configuration,determine the critical diameter
0110 C for each impact angle and velocity
0111 C
0112 DO 200 I=1,NANG
0113 C
0114 C Set the angle,in deg and radians
0115 C
0116 ANG = 0.0 + (I-1)*AINCR
0117 C
0118 C For angles > 60 deg,set ang=60
0119 C
0120 IF ( ANG .GT. 60.0 ) ANG=60.0
0121 C
0122 C Convert ang to radians
0123 C
0124 ANGR = ANG / 180.0 * 3.141592
0125 C
0126 C Set initial equal to true

```

D180-30550-4

```

0127 C
0128 C      INITIAL=.TRUE.
0129 C
0130 C      DO 100 J=1,NVEL
0131 C
0132 C      Set the velocity in ft/sec and km/sec
0133 C
0134 C      VELM = J*VINCR
0135 C
0136 C      Convert vel to ft/sec
0137 C
0138 C      VELE = VELM * 1.0E+05 / 2.54 / 12.0
0139 C
0140 C      Determine the critical diameter, as a function of wall configuration
0141 C
0142 C      IF ( CTYPE.EQ.2 ) THEN
0143 C
0144 C      For the original and regression penetration functions use DOUBLE
0145 C      subroutines
0146 C
0147 C      IF ( PFUNC.EQ.1 .OR. PFUNC.EQ.3 ) THEN
0148 C      CALL DOUBLE ( ANGR,DIA,INITIAL,ITYPE,MLI,PFUNC,SHTHK,
0149 C      1          STAND,VELE,VELM,VWTHK,BHARD,C,DENS,
0150 C      2          FSU,FTU,FY,SHPV,WILKC )
0151 C
0152 C      For Pen4 use the pen4 subroutine
0153 C
0154 C      ELSE IF ( PFUNC.EQ.2 ) THEN
0155 C      CALL PEN4 ( ANGR,BHARD,C,DENS,DIA,VWTHK,FY,FSU,FTU,
0156 C      1          INITIAL,SHATTER,SHPV,SHTHK,STAND,VELE )
0157 C      END IF
0158 C
0159 C      ELSE
0160 C      CALL SINGLE ( ANGR,DIA,VELE,VWTHK,DENS,FTU )
0161 C      END IF
0162 C
0163 C      Convert the diameter to cm
0164 C
0165 C      DIAM = DIA * 2.54
0166 C
0167 C      Store the diameter in RTABLE
0168 C
0169 C      RTABLE(J,I,PID)=DIAM
0170 C
0171 C      100    CONTINUE
0172 C
0173 C      200 CONTINUE
0174 C
0175 C      RETURN
0176 C
0177 C
0178 C      END

```

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE DOUBLE (ANGR,DIA,INITIAL,ITYPE,MLI,PFUNC,SHTHK,STAND,
0005 . VELE,VELM,VWTHK,BHARD,C,DENS,FSU,FTU,FY,
0006 . SHPV,WILKC)
0007 C
0008 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0009 C
0010 C
0011 C DOUBLE DETERMINES THE PROJECTILE DIAMETER THAT JUST PENETRATES
0012 C THE GIVEN DOUBLE PLATE CONFIGURATION AT THE GIVEN IMPACT VELOCITY
0013 C AND ANGLE. IT IS USED FOR BOTH THE ORGINAL AND REGRESSION
0014 C PENETRATION FUNCTIONS.
0015 C
0016 C
0017 C
0018 C VARIABLE LIST
0019 C
0020 C ANGR = IMPACT ANGLE MEASURED FROM THE NORMAL,RADIANS
0021 C BALL = LOGICAL PARAMETER USED TO DETERMINE IF THE BALLITIC
0022 C SUBROUTINES ARE CALLED
0023 C BHARD = ARRAY CONTAINING THE VALUES OF THE BRINNEL HARDNESS FOR
0024 C THE SHIELD & VESSEL WALL MATERIALS
0025 C BRIST = LOGICAL PARAMETER USED TO DETERMINE IF THE FRAGMENTING
0026 C SUBROUTINES ARE CALLED
0027 C C = ARRAY CONTAINING VALUES FOR THE SPEED OF SOUND (FT/SEC) FOR
0028 C THE SHIELD AND VESSEL WALL MATERIALS
0029 C DIA = PROJECTILE DIAMETER ( IN. )
0030 C DIAB = DIAMETER AS DETERMINED BY SUBROUTINE BRISTOW ( IN. )
0031 C DIABL = DIAMETER AS DETERMINED BY SUBROUTINE BALLIST ( IN. )
0032 C DIAW = DIAMETER AS DETERMINED BY SUBROUTINE WILKIN ( IN. )
0033 C EVWTHK = EQUIVALENT VESSEL WALL THK. ( IN. )
0034 C FY = ARRAY CONTAINING THE VALUES OF THE YIELD STRENGTH FOR THE
0035 C SHIELD & VESSEL WALL MATERIALS
0036 C FSU = ARRAY CONTAING VALUES OF THE ULTIMATE SHEAR STRENGTH (PSI)
0037 C FOR THE SHIELD AND VESSEL WALL MATERIALS
0038 C FTU = ARRAY CONTAING VALUES OF THE ULTIMATE TENSILE STRENGTH (PSI)
0039 C FOR THE SHIELD AND VESSEL WALL MATERIALS
0040 C INITIAL = LOGICAL PARAMETER USED TO DETERMINE IF CURRENT
0041 C SUBROUTINE CALL IS INITIAL ONE
0042 C ITYPE = ANALYSIS TYPE 1=DEBRIS & METEORIDS, 2=METEORIDS
0043 C MLI = CHARACTER STRING USED TO DETERMINE IF MLI IS USED IN WALL
0044 C PFUNC = PENETRATION FUNCTION
0045 C 1- ORIGINAL
0046 C 2- PEN4
0047 C 3- REGRESSION
0048 C SHATTER = LOGICAL PARAMETER USED TO DETERMINE IF PEN4 BALLISTIC
0049 C LIMITS HAVE BEEN EXCEEDED
0050 C SHPV = ARRAY CONTAINING THE VALUES FOR THE SHOCK PROJECTILE
0051 C VELOCITY (FT/SEC ) OF THE SHIELD & VESSEL WALL MATERIALS
0052 C SHTHK = SHIELD THICKNESS ( IN. )
0053 C STAND = SHIELD STAND-OFF DISTANCE (IN.)
0054 C THKMLI = EQUIVALENT THK OF 30 LAYERS OF MLI ( IN. )
0055 C VELE = VEL IN FT/SEC
0056 C VELM = VEL IN KM/SEC
0057 C VWTHK = VESSEL WALL THICKNESS ( IN. )
0058 C WILKC = ARRAY CONTAINING THE VALUES OF WILKINSON'S CONSTANT FOR
0059 C THE SHIELD & VESSEL WALL MATERIALS
0060 C
0061 C INTEGER*2 ITYPE,PFUNC
0062 C
0063 C DIMENSION BHARD(3),C(3),DENS(3),FY(3),FSU(3),FTU(3),SHPV(3),

```

```

0064          WILKC(3)
0065      C
0066          LOGICAL BALL,BRIST,INITIAL,MLI,SHATTER
0067      C
0068      C  INITIALIZE VARIABLES
0069      C
0070          IF ( INITIAL ) THEN
0071              BALL=.TRUE.
0072              BRIST=.TRUE.
0073              DIA=.01
0074              SHATTER=.FALSE.
0075          END IF
0076      C
0077      C  TAKE 30 LAYERS OF MLI INTO ACCOUNT USING COUR-PALAIS'S EQN.
0078      C  SET LIMIT FOR EQN. AT 10 KM/SEC OR SKIP OVER SECTION ENTIRELY
0079      C
0080      C
0081          IF (.NOT. MLI ) THEN
0082              THKMMLI = 0.0
0083              GO TO 50
0084          ENDIF
0085      C
0086          IF ( VELM .LT. 10.0 ) THEN
0087              THKMMLI = 3.045E-06 * ( VELM ** 3.42 )
0088          ELSE
0089              THKMMLI = 3.045E-06 * ( 10.0 ** 3.42 )
0090          END IF
0091      C
0092      C  CONVERT TO IN.
0093      C
0094          THKMMLI = THKMMLI / 2.54
0095      C
0096      C  ADD EQUIVALENT THK. OF MLI TO VESSEL WALL THK.
0097      C
0098      50      EVWTHK = VWTHK + THKMMLI
0099      C
0100      C
0101      C  DETERMINE DIAMETER IN IN. THAT PENETRATES THE SHIELD AND
0102      C  JUST DOES NOT PENETRATE THE VESSEL WALL
0103      C
0104      C  IF THE ANALYSIS IS FOR METEORIODS, THEN ONLY USE WILKINSON'S METHOD
0105      C  TO DETERMINE THE PENETRATION DIAMETER
0106      C
0107      C
0108          IF ( ITYPE .EQ. 2 .OR. VELM .GT. 12.0 ) THEN
0109              BALL=.FALSE.
0110              BRIST=.FALSE.
0111              GOTO 500
0112          ENDIF
0113      C
0114      C  INITIALLY CALCULATE THE PENETRATION DIAMETER USING BOTH THE
0115      C  BALLISTIC AND FRAGMENTING SUBROUTINES. THE DIAMETER CALCULATED BY
0116      C  THE BALLISTIC SUBROUTINE IS USED UNTIL THE VALUE CALCULATED FROM
0117      C  THE FRAGMENTING SUBROUTINE IS GREATER. AT THAT TIME IT IS NO LONGER
0118      C  NECESSARY TO CALL THE BALLISTIC SUBROUTINES.
0119      C
0120      C  FOR THE ORGINAL PENETRATION FUNCTION THE OLD VERSION OF PEN4 IS USED
0121      C  IN THE BALLISTIC REGIME AND THE BURCH MODIFIED EQUATIONS ARE USED
0122      C  IN THE FRAGMENTING REGIME.
0123      C
0124      C  FOR THE REGRESSION PENETRATION FUNCTION THE NEW PEN4 IS USED IN
0125      C  THE BALLISTIC REGIME AND THE REGRESSION EQUATIONS ARE USED IN
0126      C  FRAGMENTING REGIME.

```

```

0127      C                               D180-30550-4
0128      C
0129      IF (BALL) THEN
0130          IF ( PFUNC.EQ.1 ) THEN
0131              CALL BALLIST (ANGR,BHARD,C,DENS,DIA,VWTHK,FY,FSU,FTU,
0132              1              INITIAL,SHATTER,SHPV,SHTHK,STAND,VELE)
0133          ELSE
0134              CALL PEN4 ( ANGR,BHARD,C,DENS,DIA,VWTHK,FY,FSU,FTU,
0135              1              INITIAL,SHATTER,SHPV,SHTHK,STAND,VELE )
0136          END IF
0137          DIABL=DIA
0138      C
0139          IF ( PFUNC.EQ.1. ) THEN
0140              CALL BRISTOW (ANGR,C,DIA,EVWTHK,INITIAL,SHTHK,STAND,VELE)
0141          ELSE
0142              CALL REGRESS ( ANGR,DIA,MLI,SHTHK,STAND,VELM,VWTHK )
0143          END IF
0144          DIAB=DIA
0145      C
0146      C CHECK IF THE DIAMETER CALCULATED BY BALLISTIC SUBROUTINE IS LESS
0147      C THAN THAT CALCULATED BY FRAGMENTING SUBROUTINE. IF SO SET BALL TO
0148      C FALSE .
0149      C
0150          IF (DIAB.GT.DIABL) THEN
0151              BALL=.FALSE.
0152              DIA=DIAB
0153              GOTO 700
0154          ELSE
0155              BALL=.TRUE.
0156              DIA=DIABL
0157              GOTO 700
0158          ENDIF
0159      C
0160      C CALCULATE THE PENETRATION DIAMETER USING BOTH THE FRAGMENTING AND
0161      C WILKIN SUBROUTINES. THE DIAMETR CALCULATED BY FRAGMENTING IS USED
0162      C UNTIL THE VALUE DETERMINED BY WILKIN IS LESS. IT IS THEN NOT
0163      C NECESSARY TO CALL FRAGMENTING.
0164      C
0165      C
0166      ELSE
0167          IF (BRIST) THEN
0168              IF ( PFUNC.EQ.1. ) THEN
0169                  CALL BRISTOW (ANGR,C,DIA,EVWTHK,INITIAL,SHTHK,STAND,VELE)
0170              ELSE
0171                  CALL REGRESS ( ANGR,DIA,MLI,SHTHK,STAND,VELM,VWTHK )
0172              END IF
0173              DIAB=DIA
0174      C
0175              CALL WILKIN (ANGR,DENS,DIA,EVWTHK,ITYPE,SHTHK,STAND,VELM,
0176              WILKC)
0177              DIAW=DIA
0178      C
0179      C CHECK IF THE VALUE DETERMINED BY WILKIN IS LESS THEN THAT
0180      C DETERMINED BY BRISTOW. IF SO SET BRIST TO FALSE.
0181      C
0182          IF (DIAW.LT.DIAB) THEN
0183              BRIST=.FALSE.
0184              DIA=DIAW
0185              GOTO 700
0186          ELSE
0187              BRIST=.TRUE.
0188              DIA=DIAB
0189              GOTO 700

```

```
0190          ENDIF
0191      C
0192          ENDIF
0193      C
0194          ENDIF
0195      C
0196      C
0197      C   CALCULATE THE DIAMETER USING THE WILKIN SUBROUTINE.
0198      C
0199      500   CALL WILKIN (ANGR,DENS,DIA,EVWTHK,ITYPE,SHTHK,STAND,VELM,WILKC)
0200      C
0201      C   SET INITIAL TO FALSE
0202      C
0203      700   INITIAL = .FALSE.
0204      C
0205      C
0206          RETURN
0207      C
0208          END
```

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE BRISTOW (ANGR,C,DIA,EVWTHK,INITIAL,SHTHK,STAND,VELE)
0005 C
0006 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007 C
0008 C
0009 C
0010 C BRISTOW DETERMINES NUMBER OF PLATES PENETRATED. BASED ON
0011 C "MULTIPLE-DAMAGE STUDY", BY G.T. BURCH, BOEING 1967,AIR FORCE
0012 C ARMAMENT LABORATORY TECH REPORT AFATL-TR-67-116.
0013 C
0014 C VALID FROM 3.6 TO 10.2 KM/SEC,AND FOR ALUMINUM PROJECTILES
0015 C IMPACTING ALUMINUM PLATES AT NORMAL .
0016 C
0017 C
0018 C VARIABLE LIST
0019 C
0020 C ANGR = IMPACT ANGLE MEASURED FROM THE NORMAL, RADIANS
0021 C C1 = SPEED OF SOUND IN SHIELD MATERIAL,FT/SEC
0022 C C = ARRAY CONTAINING VALUES OF THE SPEED OF SOUND FOR
0023 C THE SHIELD AND VESSEL WALL MATERIALS ( FT/SEC )
0024 C CHI = INTERMEDIATE VARIABLE
0025 C DIA = PROJECTILE DIAMETER,IN.
0026 C DIA1 = PREVIOUS DIAMETER
0027 C DIA2 = CURRENT DIAMETER
0028 C DT = INTERMEDIATE VARIABLE
0029 C EVWTHK = EQUIVALENT VESSEL WALL THICKNESS,IN.
0030 C F1 = INTERMEDIATE VARIABLE
0031 C F2 = " "
0032 C F3 = " "
0033 C I = ITERATION COUNTER
0034 C INITIAL = LOGICAL PARAMETER USED TO DETERMINE IF CURRENT CALL
0035 C IS INITIAL ONE
0036 C NF = NUMBER OF PLATES PENETRATED BY THE FLIGHT PATH COMPONENT
0037 C NN = NUMBER OF PLATES PENETRATED BY THE NORMAL COMPONENT
0038 C PLP = # OF PLATES PENETRATED
0039 C PLP1 = PREVIOUS PLP
0040 C PLP2 = CURRENT PLP
0041 C SD = INTERMEDIATE VARIABLE
0042 C SHTHK = SHIELD THICKNESS,IN.
0043 C SIN3 = INTERMEDIATE VARIABLE
0044 C SLOPE = SLOPE OF LINE THROUGH (DIA1,PLP1) & (DIA2,PLP2)
0045 C SPF = NUMBER OF PLATES PENETRATED THAT ACCOUNTS FOR SPALLING
0046 C STAND = SHIELD STAND-OFF (IN.)
0047 C SWITCH = LOGICAL VARIABLE USED TO DETERMINE IF THE CRITICAL
0048 C DIAMETER IS TO BE ESTIMATED THROUGH LINEAR
0049 C INTERPOLATION
0050 C T1D = INTERMEDIATE VARIABLE
0051 C T2D = "
0052 C TEST = RATIO OF THE NUMBER OF PLATES PENETRATED TO THE
0053 C SPALLING FACTOR
0054 C VC = "
0055 C VELE = VEL IN FT/SEC
0056 C VHOLD = VELOCITY HOLDER
0057 C
0058 C
0059 C
0060 C DIMENSION C(3)
0061 C
0062 C LOGICAL INITIAL
0063 C

```

```

0064      REAL NF,NN
0065      C
0066      LOGICAL SWITCH
0067      C
0068      SAVE PLP1,PLP2,DIA1,DIA2
0069      C
0070      C
0071      C SINCE THIS METHOD DOES NOT SOLVE FOR THE DIAMETER THAT JUST
0072      C PENETRATES DIRECTLY AN ITERATIVE APPROACH IS TAKEN. INITIAL VALUES
0073      C ARE SET,AND USED TO APPROXIMATE THE CORRECT VALUE. THIS PROCESS
0074      C IS CONTINUED UNTIL THE ANSWER IS WITHIN TOLERANCES.
0075      C
0076      C
0077      C SET INITIAL VALUES
0078      C
0079      I=0
0080      SPF = 0.85
0081      SWITCH=.FALSE.
0082      C
0083      C IF THIS IS THE INITIAL CALL SET INITIAL ELSE USE PREVIOUS VALUES
0084      C VALUES
0085      C
0086      IF (INITIAL) THEN
0087          DIA2=10.0
0088      ENDIF
0089      C
0090      C FOR VELOCITIES LESS THAN 11,800 FT/SEC SET THE VELOCITY EQUAL
0091      C TO 11,800 AND CALCULATE THE CRITICAL DIAMETER. USE THIS VALUE
0092      C TO ESTIMATE THE ACTUAL DIAMETER. SET THE LOGICAL VARIABLE SWITCH
0093      C TO TRUE AND SAVE THE VELOCITY AS VHOLD.
0094      C
0095      IF ( VELE .LT. 11800. ) THEN
0096          VHOLD=VELE
0097          VELE=11800.
0098          SWITCH=.TRUE.
0099      END IF
0100      C
0101      C CALCULATE INTERMEDIATE VARIABLES THAT DO NOT NEED TO BE CALCULATED
0102      C FOR EACH DIAMETER
0103      C
0104      CHI=TAN(ANGR)-0.50
0105      SIN3=(SIN(ANGR))**3
0106      VC=VELE/C(1)
0107      C
0108      100 I=I+1
0109      C
0110      C IF THIS IS THE FIRST PREDICTION USE THE DIAMETER VALUE THAT WAS
0111      C CALCULATED TO PENETRATE FOR THE PREVIOUS CASE AS A STARTING POINT.
0112      C ELSE USE A LINEAR PREDICTION APPROACH BASED ON THE LAST TWO
0113      C PREDICTIONS.
0114      C
0115      IF ( I.EQ.1 ) THEN
0116      C
0117          DIA=DIA2
0118          DIA2=0.0
0119          PLP2=0.0
0120      C
0121      ELSE
0122      C
0123          SLOPE=(PLP2-PLP1)/(DIA2-DIA1)
0124          DIA=((SPF-PLP1)/SLOPE+DIA1)
0125      C
0126      C CHECK THAT DIA > 0.0

```

```

0127 C
0128 IF (DIA.LT.0.0) DIA=1.0E-10
0129 C
0130 C Check if dial = dia2,if so stop
0131 C
0132 IF ( DIA.EQ.DIA2 ) THEN
0133 WRITE ( 6,150 ) ANGR,VELE,DIA
0134 150 FORMAT (/1X,'BRISTOW CANNOT CONVERGE BECAUSE OF FLAT ',
0135 1 'SLOPE (ANGLE,VEL,DIA) =',3E12.5 )
0136 STOP
0137 END IF
0138 END IF
0139 C
0140 C CALCULATE # OF PLATES PENETRATED
0141 C
0142 SD=STAND/DIA
0143 T1D=SHTHK/DIA
0144 T2D=EVWTHK/DIA
0145 C
0146 F1=2.42*(T1D**(-1./3.))+4.26*(T1D**(1./3.))-4.18
0147 C
0148 F2=(0.5-1.87*T1D)+(5.*T1D-1.6)*(CHI**3)+(1.7-12.*T1D)*CHI
0149 C
0150 F3=0.32*(T1D**(5./6.))+0.48*(T1D**(1./3.))*SIN3
0151 C
0152 NF=(F1+0.63*F2)*(VC**(-4./3.))*(SD**(-5./12.))*(T2D**(-7./12.))
0153 C
0154 NN=F3*(T2D**(-1.))*(VC**(-4./3.))
0155 C
0156 C DETERMINE WHICH COMPONENT CONTROLS
0157 C
0158 IF ( NF.GT.NN ) THEN
0159 PLP=NF
0160 ELSE
0161 PLP=NN
0162 ENDIF
0163 C
0164 C RESET HOLDERS
0165 C
0166 DIA1=DIA2
0167 DIA2=DIA
0168 PLP1=PLP2
0169 PLP2=PLP
0170 C
0171 C CHECK IF PLP IS WITHIN TOLERANCE,IF NOT ITERATE
0172 C
0173 TEST = PLP/SPF
0174 C
0175 C
0176 IF ( TEST.LT.0.99 .OR. TEST.GT.1.01 ) GO TO 100
0177 C
0178 C IF SWITCH IS TRUE, ESTIMATE THE CRITICAL DIAMETER USING LINEAR
0179 C INTERPOLATION. THE TWO POINTS USED ARE THE ORIGIN AND THE VALUE
0180 C CALCULATED AT 11800 .
0181 C
0182 IF ( SWITCH ) THEN
0183 C
0184 DIA=DIA2*VHOLD/11800.0
0185 SWITCH=.FALSE.
0186 DIA1=0.0
0187 PLP1=0.0
0188 C
0189 END IF

```

D180-30550-4

0190	C	
0191		RETURN
0192	C	
0193	C	
0194		END

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE WILKIN (ANGR,DENS,DIA,EVWTHK,ITYPE,SHTHK,STAND,VELM,
0005 WILKC )
0006 C
0007 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008 C
0009 C WILKIN DETERMINES THE DIAMETER THAT JUST PENETRATES THE VESSEL
0010 C WALL. IT IS BASED ON J.P.D. WILKINSON'S PAPER 'A PENETRATION
0011 C CRITERION FOR DOUBLED-WALLED STRUCTURE SUBJECT TO METEOROID IMPACT'
0012 C ,AIAA JOURNAL OCT 1969.
0013 C
0014 C THE MAJOR ASSUMPTION USED BY WILKINSON IS THAT THE PROJECTILE
0015 C VAPORIZES ON IMPACT
0016 C
0017 C
0018 C VARIABLE LIST
0019 C
0020 C ANGR = IMPACT ANGLE MEASURED FROM THE NORMAL, RADIANS
0021 C CONST = MATERIAL CONSTANT DEFINED BY WILKINSON
0022 C C1 = INTERMEDIATE VARIABLE
0023 C DENS = ARRAY CONTAINING VALUES FOR DENSITY OF THE SHIELD &
0024 C VESSEL WALL MATERIALS
0025 C DIA = PROJECTILE DIAMETER (IN)
0026 C DIAM = PROJECTILE DIAMETER (CM )
0027 C EVWTHK = EQUIVALENT VESSEL WALL THK. ( IN)
0028 C ITYPE = ANALYSIS TYPE 1=DEBRIS & METEORIDS, 2=METEORIDS
0029 C MASS = PROJECTILE MASS ( GRAMS )
0030 C MB = INTERMEDIATE VARIABLE
0031 C MT = " "
0032 C PI = PI
0033 C PROJDEN = PROJECTILE DENSITY ( G/CC )
0034 C RMI = UNIT MASS OF SHIELD (G/CM**2)
0035 C RM2 = UNIT MASS OF VESSEL WALL (G/CM**2)
0036 C SHDEN = SHIELD & VESSEL WALL DENISITY ( GRAMS/CC )
0037 C SHTHK = SHIELD THK. (IN)
0038 C STAND = SHIELD STAND-OFF DISTANCE (IN)
0039 C STANDM = STAND IN CM.
0040 C VELM = VEL IN KM/SEC
0041 C VNORM = NORMAL COMPONENT OF THE VELEOCITY VECTOR,KM/SEC
0042 C VWDEN = VESSEL WALL DENSITY (LB/IN**3)
0043 C WILKC = ARRAY CONTAINING VALUES OF WILKINSON'S CONSTANT
0044 C FOR THE SHIELD & VESSEL WALL MATERIALS
0045 C
0046 C
0047 C
0048 C DIMENSION DENS(3),WILKC(3)
0049 C
0050 C INTEGER*2 ITYPE
0051 C
0052 C REAL MB,MT,MASS
0053 C
0054 C SET INITIAL VALUES
0055 C
0056 C PI = 3.141592654
0057 C
0058 C SET PROJECTILE DENSITY IN G/CC
0059 C
0060 C PROJDEN = DENS(3)*27.705
0061 C
0062 C SET SHIELD AND VESSEL WALL DENSITY IN G/CC
0063 C

```

D180-30550-4

```

0064      SHDEN = DENS(1) * 27.705
0065      VWDEN = DENS(2) * 27.705
0066      C
0067      C  CONST IS A MATERIAL VARIABLE DEFINED IN THE PAPER & IS
0068      C  EQUAL TO .401 FOR 2219 AL
0069      C
0070      CONST = WILKC(2)
0071      C
0072      C  DETERMINE SHIELD & VESSEL WALL MASS PER UNIT AREA
0073      C
0074      RM1 = SHTHK * 2.54 * SHDEN
0075      RM2 = EVWTHK * 2.54 * VWDEN
0076      C
0077      C  CONVERT STAND TO CM
0078      C
0079      STANDM = STAND * 2.54
0080      C
0081      C  CALCULATE THE NORMAL COMPONENT OF THE VELOCITY VECTOR
0082      C
0083      VNORM=VELM*COS(ANGR)
0084      C
0085      C  DETERMINE CRITICAL PROJECTILE DIAMETER
0086      C
0087      MT = 1.44*(PI/6.0)**(1./3.)*CONST*RM1*RM2*STANDM**2.0
0088      MB = PROJDN**(2./3.)*VNORM
0089      MASS = (MT/MB)**.75
0090      DIAM = (6.0*MASS/(PI*PROJDN))**(1./3.)
0091      C
0092      C  CHECK IF APPROPRIATE EQ. WAS USED
0093      C
0094      C1 = RM1 / ( PROJDN * DIAM )
0095      IF ( C1 .GT. 1.0 ) THEN
0096      C
0097      C  WRONG EQN. USED,RECALC USING CORRECT EQN.
0098      C
0099      MASS= 1.44*CONST*RM2*(STANDM**2.)/VNORM
0100      DIAM = (6.0*MASS/(PI*PROJDN))**(1./3.)
0101      END IF
0102      C
0103      C  CONVERT DIAMETER TO IN
0104      C
0105      DIA = DIAM / 2.54
0106      C
0107      RETURN
0108      C
0109      END

```

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE REGRESS ( ANGR,DIA,MLI,SHTHK,STAND,VELM,VWTHK )
0005 C
0006 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0007 C
0008 C
0009 C
0010 C Subroutine REGRESS determines the critical projectile diameter for
0011 C a two-plate structure impacted by aluminum spherical projectiles. It
0012 C was developed from test data obtained during the NASA contract
0013 C 'Integrated Space Station Wall Design Guide and Penetration Control
0014 C Plan'. The data varied from 2 to 8 km/sec.
0015 C
0016 C
0017 C Variable List
0018 C
0019 C angr = impact angle measured from the normal, radians
0020 C dia = critical projectile diameter, in
0021 C dl3 = intermediate variable
0022 C high = holder, last diameter to penetrate , in
0023 C ic = counter
0024 C low = holder, last diameter to not penetrate, in
0025 C lsd = intermediate variable
0026 C mli = logical variable indicating wether multi-layer insulation is
0027 C used
0028 C mt = intermediate variable
0029 C np = number of plates penetrated, excluding the shield
0030 C shthk = shield thickness , in
0031 C stand = shield stand-off , in
0032 C switch = logical variable, used to determine if a penetrating
0033 C diameter has been determined
0034 C tt = intermediate variable
0035 C t13 = " "
0036 C vc2 = " "
0037 C velm = impact velocity, km/sec
0038 C vwthk = vessel wall thickness , in
0039 C
0040 C
0041 C
0042 C INTEGER IC
0043 C
0044 C REAL LOW,LSD,MT,NP
0045 C
0046 C LOGICAL MLI,SWITCH
0047 C
0048 C Intialize Variables
0049 C
0050 C IC=0
0051 C LOW=0.0
0052 C HIGH=5.0
0053 C SWITCH=.TRUE.
0054 C
0055 C Set MLI constant
0056 C
0057 C IF ( MLI ) THEN
0058 C MT=-14.0
0059 C ELSE
0060 C MT=0.0
0061 C END IF
0062 C
0063 C Since the equation does not solve for the critical projectile

```

D180-30550-4

```

0064 C diameter directly, use a binary search technique to determine it.
0065 C First determine a diameter that penetrates then narrow in on the
0066 C actual diameter.
0067 C
0068 100 CONTINUE
0069 C
0070     IC=IC+1
0071 C
0072     IF ( SWITCH ) THEN
0073         HIGH=HIGH*2.0
0074     END IF
0075 C
0076     DIA=(HIGH+LOW)/2.0
0077 C
0078 C Check that the diameter is less then 50 cm , if not stop
0079 C
0080     IF ( DIA.GT.50.0 ) THEN
0081         WRITE ( 6, ' (''---ERROR--- Diameter greater than 50 cm in '',
0082 1         'subroutine REGRESS'')' )
0083         STOP
0084     END IF
0085 C
0086 C Calculate the intermediate variables
0087 C
0088     D13=DIA**(1./3.)
0089     LSD=LOG10(STAND)/DIA
0090     T13=SHTHK**(1./3.)
0091     TT=TAN(ANGR)
0092     VC2=VELM*(COS(ANGR)**2)
0093 C
0094 C Calculate the number of plates penetrated
0095 C
0096     NP=1.52-6.18*T13-18.8*VWTHK-0.146*LSD+MT*SHTHK+10.8*D13-0.287*VC2
0097 1     -0.713*TT
0098 C
0099 C Check for convergance
0100 C
0101     IF ( IC.GT.100 ) THEN
0102         WRITE ( 6, ' (''---ERROR--- REGRESS failed to converge after '',
0103 1         '100 cycles'')' )
0104         STOP
0105     END IF
0106 C
0107 C Has a diameter that penetrates been found, if not reset holders and
0108 C try again
0109 C
0110     IF ( SWITCH ) THEN
0111         IF ( NP.GT.1 ) THEN
0112             SWITCH=.FALSE.
0113             HIGH=DIA
0114             GO TO 100
0115         END IF
0116     END IF
0117 C
0118 C Does the diameter yeild an acceptable result, if not rest holders
0119 C and try again
0120 C
0121     IF ( NP.LT.0.999 ) THEN
0122         LOW=DIA
0123         GO TO 100
0124     ELSE IF ( NP.GT.1.001 ) THEN
0125         HIGH=DIA
0126         GO TO 100

```

```
0127          END IF
0128      C
0129      C  Finished
0130      C
0131          RETURN
0132      C
0133          END
```

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE BALLIST (ANGR,BHARD,C,DENS,DIA,VWTHK,FY,FSU,FTU,
0005 1 INITIAL,SHATTER,SHPV,SHTHK,STAND,VELE )
0006 C
0007 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008 C
0009 C BALLISTIC DETERMINES THE DIAMETER THAT JUST PENETRATES. IT ULTILZES
0010 C THE BALLISTIC PORTION OF BOEING'S HYPERVELOCITY CODE PEN4.
0011 C
0012 C
0013 C VARIABLE LIST
0014 C
0015 C ANGR = IMPACT ANGLE MEASURED FROM THE NORMAL,RADIANS
0016 C BHARD = ARRAY CONTAINING THE VALUES OF THE BRINNEL HARDNESS FOR
0017 C THE SHIELD & VESSEL WALL MATERIALS
0018 C C = ARRAY CONTAINING VALUES FOR THE SPEED OF SOUND FOR THE
0019 C SHIELD AND VESSEL WALL MATERIALS (FT/SEC)
0020 C DEN = PROJECTILE DENSITY (LB/CFT)
0021 C DENS = ARRAY CONTAINING VALUES FOR THE DENSITY FOR THE
0022 C SHIELD AND VESSEL WALL MATERIALS (LB/IN**3)
0023 C DIA = SPHERICAL PROJECTILE DIAMETER (IN)
0024 C DIA1 = PREVIOUS DIAMETER THAT PENETRATED (IN)
0025 C DIA2 = DIAMETER THAT PENETRATED TWO ITERATIONS PREVIOUSLY (IN)
0026 C EVWTHK = EQUIVALENT VESSEL WALL THICKNESS,IN.
0027 C FY = ARRAY CONTAINING THE VALUES OF THE YIELD STRENGTH FOR THE
0028 C SHIELD & VESSEL WALL MATERIALS
0029 C FSU = ARRAY CONTAINING VALUES OF THE ULTIMATE SHEAR STRENGTH
0030 C FOR THE SHIELD AND VESSEL WALL MATERIALS (PSI)
0031 C FTU = ARRAY CONTAINING VALUES OF THE ULTIMATE TENSILE STRENGTH
0032 C FOR THE SHIELD AND VESSEL WALL MATERIALS (PSI)
0033 C I = COUNTER
0034 C INITIAL = LOGICAL PARAMETER USED TO DETRMINIE IF CURRENT CALL
0035 C IS INITIAL ONE.
0036 C MASS = PROJECTILE MASS,LBS
0037 C N = INCREMENT MULIPLIER
0038 C PDENS = DENS ARRAY CONVERTED TO SLUGS/FT**3
0039 C PFY = FY ARRAY CONVERTED TO PSF
0040 C PFSU = FSU ARRAY CONVERTED TO PSF
0041 C PFTU = FTU ARRAY CONVERTED TO PSF
0042 C P1 = LAST MASS GUESS TO NOT PENETRATE
0043 C P2 = LAST MASS GUESS TO PENETRATE
0044 C PEN = TRUE OR FALSE
0045 C PI = 3.14
0046 C PMINCR = INITIAL MASS GUESS INCREMENT
0047 C RATIO = P2 / P1
0048 C SHATTER = LOGICAL PARAMETER USED TO DETERMINE IF PEN4 BALLISTIC
0049 C LIMITS HAVE BEEN EXCEEDED
0050 C SHPV = ARRAY CONTAING VALUES OF THE SHOCK PROJECTILE VELOCITY
0051 C FOR THE SHIELD AND VESSEL WALL MATERIALS,( UNITLESS )
0052 C SHTHK = SHIELD THICKNESS,IN.
0053 C SPACE = ARRAY CONTAINING THE SHIELD SPACING ,FT.
0054 C STAND = SHIELD STAND-OFF,IN.
0055 C TARMAT = ARRAY CONTAINING MATERIAL PROPERTY POINTERS
0056 C THETA = IMPACT ANGLE (RAD),MEASURED FROM THE NORMAL
0057 C THICK = SHIELD & VESSEL WALL THICKNESS,FT.
0058 C VELE = COLLISION VELOCITY,FT/SEC
0059 C VEL1 = VEL FOR DIA1
0060 C VEL2 = VEL FOR DIA2
0061 C V0 = IMPACT VELOCITY (FT/SEC)
0062 C VR2 = RESIDUAL VELOCITY AFTER VESSEL WALL,FT/SEC
0063 C

```

```

0064      C                               D180-30550-4
0065      C
0066      DIMENSION BHARD(3),C(3),DENS(3),FY(3),FSU(3),FTU(3),PDENS(3),
0067      .          PFY(3),PFSU(3),PFTU(3),SHPV(3),SPACE(2),THICK(3),
0068      .          TARMAT(3)
0069      C
0070      INTEGER PROJMAT,TARMAT
0071      C
0072      LOGICAL PEN,INITIAL,SHATTER
0073      C
0074      REAL MASS
0075      C
0076      SAVE DIA1,DIA2,P2,VEL1,VEL2
0077      C
0078      C
0079      C SET INITIAL VALUES
0080      C
0081      I = 0
0082      N = 0
0083      PEN = .FALSE.
0084      PMINCR = 1.0E-04
0085      PI = 3.1415926536
0086      P1 = 1.0E-06
0087      C
0088      C CONVERT VARIABLES TO PEN4 FORMAT AND UNITS
0089      C
0090      DO 50 I=1,3
0091      C
0092      C CONVERT DENS TO SLUGS/FT**3
0093      C
0094      PDENS(I)=DENS(I)*1728./32.2
0095      C
0096      C CONVERT FSU AND FTU TO PSF
0097      C
0098      PFY(I)=FY(I)*144.
0099      PFTU(I)=FTU(I)*144.
0100      PFSU(I)=FSU(I)*144.
0101      C
0102      50 CONTINUE
0103      C
0104      PROJMAT = 3
0105      SPACE(1) = STAND/12.0
0106      SPACE(2) = 0.0
0107      TARMAT(1) = 1
0108      TARMAT(2) = 2
0109      TARMAT(3) = 0
0110      THETA = ANGR
0111      THICK(1) = SHTHK/12.0
0112      THICK(2) = VWTHK/12.0
0113      THICK(3) = 0.0
0114      V0 = VELE
0115      C
0116      C PEN4 DOES NOT SOLVE FOR THE CRITICAL PROJECTILE DIAMETER DIRECTLY
0117      C THEREFORE AN ITERATIVE APPROACH IS TAKEN. PEN4 ONLY RETURNS
0118      C A LOGICAL PARAMETER INDICATING IF PENETRATION HAS OCCURED.
0119      C
0120      C
0121      C FOR INITIAL CASE SET P2 > 0.0
0122      C
0123      IF (INITIAL) THEN
0124      P2 = 1.0E-04
0125      ENDIF
0126      C

```

D180-30550-4

```

0127 C IF SHATTER HAS OCCURED NO NEED TO CALL PEN4,USE LINEAR APPROX.
0128 C
0129 IF (SHATTER) GO TO 500
0130 C
0131 C DETERMINE INITIAL MASS THAT PENETRATES
0132 C
0133 100 MASS = P2 + N * PMINCR
0134 C
0135 NC = NC + 1
0136 C
0137 CALL OPEN4 (PEN,MASS,THETA,SHATTER,V0,PDENS,BHARD,PROJMAT,SHPV,
0138 C,SPACE,TARMAT,THICK,PFY,*500)
0139 C
0140 IF ( PEN ) THEN
0141 C
0142 C IF PENETRATION TRY WITH MASS BETWEEN P1 & P2
0143 C
0144 PEN = .FALSE.
0145 P2 = MASS
0146 MASS = ( P1 + P2 ) / 2.0
0147 C
0148 NC = NC + 1
0149 CALL OPEN4 (PEN,MASS,THETA,SHATTER,V0,PDENS,BHARD,PROJMAT,
0150 SHPV,C,SPACE,TARMAT,THICK,PFY,*500)
0151 C
0152 C
0153 200 IF ( PEN ) THEN
0154 C
0155 C IF PENETRATION,SET P2 = MASS & CHECK RATIO
0156 C
0157 PEN = .FALSE.
0158 P2 = MASS
0159 RATIO = P2 /P1
0160 IF ( RATIO .LT. .995 .OR. RATIO .GT. 1.005 ) THEN
0161 C
0162 C IF TRUE TRY AGAIN BETWEEN P1 & P2
0163 C
0164 MASS = ( P1 + P2 ) / 2.0
0165 NC = NC + 1
0166 CALL OPEN4 (PEN,MASS,THETA,SHATTER,V0,PDENS,BHARD,
0167 PROJMAT,SHPV,C,SPACE,TARMAT,THICK,PFY,
0168 *500)
0169 GO TO 200
0170 ELSE
0171 C
0172 C IF FALSE,MASS IS WITHIN RANGE SO CONTINUE
0173 C
0174 GO TO 300
0175 END IF
0176 C
0177 ELSE
0178 C
0179 C IF FALSE,SET P1 = MASS AND CHECK RATIO
0180 C
0181 P1 = MASS
0182 RATIO = P2 / P1
0183 IF ( RATIO .LT. .995 .OR. RATIO .GT. 1.005 ) THEN
0184 C
0185 C IF TRUE,TRY AGAIN BETWEEN P1 & P2
0186 C
0187 MASS = ( P1 + P2 ) / 2.0
0188 NC = NC + 1
0189 CALL OPEN4 (PEN,MASS,THETA,SHATTER,V0,PDENS,BHARD,

```

D180-30550-4

```

0190      .      PROJMAT, SHPV, C, SPACE, TARMAT, THICK, PFY,
0191      .      *500)
0192      GO TO 200
0193      ELSE
0194      C
0195      C      IF FALSE, MASS IS WITHIN RANGE SO CONTINUE
0196      C
0197      GO TO 300
0198      END IF
0199      ENDIF
0200      ELSE
0201      C
0202      C      IF FALSE INCREASE INCREMENT & TRY AGAIN
0203      C
0204      P1 = MASS
0205      I = I + 1
0206      N = 2 ** I
0207      GO TO 100
0208      END IF
0209      C
0210      P2=MASS
0211      C
0212      C      CALCULATE DIAMETER
0213      C
0214      C
0215      C
0216      300      DEN = DENS(3)*1728.0
0217      C
0218      DIA = ((MASS*6.0)/(PI*DEN))**(1.0/3.0)
0219      C
0220      C      CONVERT TO IN.
0221      C
0222      DIA = DIA * 12.0
0223      C
0224      C
0225      C
0226      C      RESET HOLDERS
0227      C
0228      DIA2=DIA1
0229      DIA1=DIA
0230      VEL2=VEL1
0231      VEL1=VELE
0232      C
0233      C
0234      RETURN
0235      C
0236      C      SINCE SHATTER HAS OCCURED NO NEED TO CALL PEN4 USE LINEAR APPROXL
0237      C
0238      500      SLOPE=(DIA1-DIA2)/(VEL1-VEL2)
0239      DIA=DIA2+SLOPE*(VELE-VEL2)
0240      RETURN
0241      C
0242      END

```

D180-30550-4

```

0001 C
0002 SUBROUTINE OPEN4 (Penetration,Mass,Theta,SHATTER,V0,
0003 1 Density,Hardness,ProjMat,ShockProjVel,SoundVel,
0004 2 Spacing,TarMat,Thick,YieldStrength,*)
0005 C This version of pen4 is strictly for use in the sub-shatter velocity
0006 C regime. It is a modified version of spin14, created on 6/21/85.
0007
0008 LOGICAL Penetration,SHATTER
0009 REAL Mass,LAMBDA,NR,NF,NH,J,NR2,NHT,LastPK,MR
0010 INTEGER ProjMat,TarMat,Exponent,TopCount,BottomCount1,BottomCount2
0011 DIMENSION Thick(3),TarMat(3),Spacing(2),
0012 : Density(3),YieldStrength(3),SoundVel(3),
0013 : ShockProjVel(3),Hardness(3),Epsil(2),
0014 : NR(3),NR2(3),RF(3),Flagit(3)
0015 C
0016 C Density in Slugs/CubicFoot
0017 C YieldStrength in Lbs/SquareFoot
0018 C SoundVel in Feet/Second
0019 C Hardness is Brinell Scale
0020 DATA Epsil,Gamma/5.71,5.71,90.0/
0021 DATA F1,F2/4.0,1.0/
0022 DATA EffectiveThicknessRatio/.6/
0023 DATA A,B/2.0,3.125E-04/
0024 PI=3.141592653589793
0025 Small = 1E-36
0026 RecipSqrt2PI=1./SQRT(2.*PI)
0027 C *****Calculate Radius of Projectile Sphere*****
0028 RP=(Mass*3./
0029 1 (Density(ProjMat)*32.2*4.*PI))**(1./3.)
0030 Diam = RP * 2.
0031 C *****
0032
0033 C *****Compute ResidualVel*****
0034 VDEL=( V0**2*1.33*RP**2*Density(ProjMat)-
0035 1 F1*YieldStrength(TarMat(1))*THICK(1)**2/COS(THETA)**2
0036 2 *A*EXP(-B*V0) )
0037 2 /( 1.33*RP**2*Density(ProjMat)+F2*RP*THICK(1)*
0038 3 Density(TarMat(1))/COS(THETA) )
0039 VR=SQRT(AMAX1(VDEL,0.))
0040 C *****
0041
0042 C *****Cratering V50*****
0043 V50=SQRT((Thick(1)*EffectiveThicknessRatio/COS(Theta)/
0044 1 (0.281*Diam*(Density(ProjMat)/Density(TarMat(1)))**(1./3.)))
0045 2 *(1./31)/Density(ProjMat)*(2.*YieldStrength(TarMat(1))))
0046 C *****
0047
0048 C *****IF No Penetration Report Result and Exit*****
0049 IF (V0.LT.V50) THEN
0050 Penetration = .FALSE.
0051 RETURN
0052 ELSE IF (Thick(2).EQ.0.0) THEN
0053 Penetration = .TRUE.
0054 RETURN
0055 ENDIF
0056 C *****
0057
0058 C *****Mass Loss Regime Decision*****
0059 ToverD = Thick(1)/Diam
0060 FTHETA=-2.433E-4 - 1.643E-2*COS(THETA)+3.201*COS(THETA)**2-
0061 : 2.184*COS(THETA)**3
0062 IF (ToverD.LT..40) THEN
0063 Vf = 4100.

```

```

0064      ELSE
0065          Vf = 4986.*ToverD**.21
0066      END IF
0067      IF (V0.LT.VF+4000.) THEN
0068          IF (V0.LT.Vf) THEN
0069              CALL MassErr(V0,Mass,Hardness(ProjMat),
0070              :           Density(ProjMat),Density(TarMat(1)),Thick(1),Diam,MR)
0071              ELSE
0072                  CALL Fract(MR,V0,Vf,FTheta,Mass,ToverD)
0073              END IF
0074              CALL LarMR(VR,MR,V50,Thick(2),EffectiveThickNessRatio,Theta,
0075              :           Diam,Density(ProjMat),Density(TarMat(2)),
0076              :           YieldStrength(TarMat(1)),Penetration)
0077          ELSE
0078              C      *****The rest of this subroutine contains the evaluation*****
0079              C      ***** of shatter regime multiple cratering penetration*****
0080              C      #####Shatter regime removed for independent use of ballistic#####
0081              C      #####And fracture regime evaluation#####
0082              SHATTER = .TRUE.
0083              RETURN 1
0084          END IF
0085          RETURN
0086      END

```

D180-30550-4

```

0001
0002      SUBROUTINE Fract (MR,V0,Vf,FTheta,MProj,ToverD)
0003      REAL MR,MProj
0004      IF (V0.GT.Vf+2000.) THEN
0005          MR = MProj*FTheta*.16
0006      ELSE IF (ToverD.GT..25) THEN
0007          MR = MProj*FTheta*.25
0008      ELSE
0009          MR = MProj*FTheta*.667
0010      END IF
0011      RETURN
0012      END
0001
0002      SUBROUTINE LarMR (VR,MR,V50,Thickness,
0003      :                   EffectiveThicknessRatio,Theta,Diam,ProjDensity,
0004      :                   TargetDensity,TargetYieldStrength,Penetration)
0005      REAL MR
0006      LOGICAL Penetration
0007      V50=SQRT((Thickness*EffectiveThicknessRatio/COS(Theta)/
0008      1  (0.281*Diam*(ProjDensity/TargetDensity)**.33333))
0009      2  **(1./31)/ProjDensity*(2.*TargetYieldStrength))
0010      IF (VR.LT.V50) THEN
0011          Penetration = .FALSE.
0012      ELSE
0013          Penetration = .TRUE.
0014      END IF
0015      RETURN
0016      END
0001
0002      SUBROUTINE MassErr (V1,MFC1,BHNC,RhoP,RhoT,Thick,Diam,MR)
0003      REAL MR,MS1,MFC1
0004      RhoC = RhoP*.01873
0005      RhoS = RhoT*.01873
0006      T = Thick*12.
0007      DFC1 = Diam*12
0008      C      DFC1 = SQRT(4*AC/PI)
0009      UFC=SQRT(2680*BHNC/RHOC)
0010      VP=V1/UFC
0011      AC1=PI*DFC1**2/(4*COS(THETA))
0012      MS1=RHOS*T*AC1
0013      MR=MFC1+0.5*MS1*LOG(2.74/(1+VP**2))
0014      MR = AMIN1(MR,MFC1)
0015      RETURN
0016      END

```

```

0001      C                               D180-30550-4
0002      C
0003      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0004      C
0005      C          SUBROUTINE SINGLE ( ANGR,DIA,VELE,VWTHK,DENS,FTU )
0006      C
0007      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008      C
0009      C
0010      C      Single determines the critical projectile diameter for single plate
0011      C      configurations. It is based on the Schmidt-Holsapple crater volume
0012      C      equation. The equation was solved for the critical diameter as a
0013      C      function of the plate and projectile properties in english units.
0014      C      It assumes that a hemispherical crater depth equal to 70 percent of
0015      C      the plate thickness causes failure. This is a attempt to account for
0016      C      spall and was reccomended by Mike Bjorkman of the Boeing SHock Physics
0017      C      group.
0018      C
0019      C
0020      C      Ref: 'On Scaling of the Crater Dimensions 2. Impact Processes',
0021      C      K.A. Holsapple & R.M. Schmidt, JGeophy Res, v87,nb3,10 March 1982,
0022      C      p1849-70
0023      C
0024      C
0025      C      Variable List
0026      C
0027      C      angr = impact angle measured from the normal, radians
0028      C      dia = critical projectile diameter, in
0029      C      dr = intermediate variable
0030      C      fr =      "      "
0031      C
0032      C      vele = relative velocity of the projectile, ft/sec
0033      C      vwthk = vessel wall thickness, in
0034      C
0035      C      Array list
0036      C
0037      C      dens = density, lb/in**3
0038      C              1- shield
0039      C              2- vessel wall
0040      C              3- projectile
0041      C      ftu = ultimate tensile strength, psi
0042      C              1- shield
0043      C              2- vessel wall
0044      C              3- projectile
0045      C
0046      C
0047      C      DIMENSION DENS(3),FTU(3)
0048      C
0049      C      DR=(DENS(3)/DENS(2))**(-0.159)
0050      C      V2=(VELE*COS(ANGR))**2
0051      C      FR=(2.6833*FTU(2)/DENS(3)/V2)**0.236
0052      C
0053      C      DIA=1.442*DR*FR*VWTHK
0054      C
0055      C      RETURN
0056      C
0057      C      END

```

D180-30550-4

```

0001 C
0002 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003 C
0004 SUBROUTINE PEN4 ( ANGR,BHARD,C,DENS,DIA,VWTHK,FY,FSU,FTU,
0005 1 INITIAL,SHATTER,SHPV,SHTHK,STAND,VELE )
0006 C
0007 CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008 C
0009 C PEN4 DETERMINES THE DIAMETER THAT JUST PENETRATES. IT UTILIZES
0010 C THE UPDATED BOEING'S HYPERVELOCITY CODE PEN4 VERSION 10.
0011 C
0012 C
0013 C VARIABLE LIST
0014 C
0015 C ANGR = IMPACT ANGLE MEASURED FROM THE NORMAL , RADIANS
0016 C BHARD = ARRAY CONTAINING THE VALUES OF THE BRINNEL HARDNESS FOR
0017 C THE SHIELD & VESSEL WALL MATERIALS
0018 C C = ARRAY CONTAINING VALUES FOR THE SPEED OF SOUND FOR THE
0019 C SHIELD AND VESSEL WALL MATERIALS (FT/SEC)
0020 C DEN = PROJECTILE DENSITY (LB/CFT)
0021 C DENS = ARRAY CONTAINING VALUES FOR THE DENSITY FOR THE
0022 C SHIELD AND VESSEL WALL MATERIALS (LB/IN**3)
0023 C DIA = SPHERICAL PROJECTILE DIAMETER (IN)
0024 C DIA1 = PREVIOUS DIAMETER THAT PENETRATED (IN)
0025 C DIA2 = DIAMETER THAT PENETRATED TWO ITERATIONS PREVIOUSLY (IN)
0026 C EVWTHK = EQUIVALENT VESSEL WALL THICKNESS , IN.
0027 C FY = ARRAY CONTAINING THE VALUES OF THE YIELD STRENGTH FOR THE
0028 C SHIELD & VESSEL WALL MATERIALS
0029 C FSU = ARRAY CONTAINING VALUES OF THE ULTIMATE SHEAR STRENGTH
0030 C FOR THE SHIELD AND VESSEL WALL MATERIALS (PSI)
0031 C FTU = ARRAY CONTAINING VALUES OF THE ULTIMATE TENSILE STRENGTH
0032 C FOR THE SHIELD AND VESSEL WALL MATERIALS (PSI)
0033 C I = COUNTER
0034 C INITIAL = LOGICAL PARAMETER USED TO DETRMINIE IF CURRENT CALL
0035 C IS INITIAL ONE.
0036 C MASS = PROJECTILE MASS , LBS
0037 C N = INCREMENT MULIPLIER
0038 C P1 = LAST MASS GUESS TO NOT PENETRATE
0039 C P2 = LAST MASS GUESS TO PENETRATE
0040 C PEN = TRUE OR FALSE
0041 C PI = 3.14
0042 C PMINCR = INITIAL MASS GUESS INCREMENT
0043 C RATIO = P2 / P1
0044 C SHATTER = LOGICAL PARAMETER USED TO DETERMINE IF PEN4 BALLISTIC
0045 C LIMITS HAVE BEEN EXCEEDED
0046 C SHPV = ARRAY CONTAING VALUES OF THE SHOCK PROJECTILE VELOCITY
0047 C FOR THE SHIELD AND VESSEL WALL MATERIALS , ( UNITLESS )
0048 C SHTHK = SHIELD THICKNESS , IN.
0049 C SPACE = ARRAY CONTAINING THE SHIELD SPACING ,FT.
0050 C STAND = SHIELD STAND-OFF , IN.
0051 C TARMAT = ARRAY CONTAINING MATERIAL PROPERTY POINTERS
0052 C THETA = IMPACT ANGLE (DEG) , MEASURED FROM THE NORMAL
0053 C THICK = SHIELD & VESSEL WALL THICKNESS , FT.
0054 C VELE = COLLISION VELOCITY , FT/SEC
0055 C VEL1 = VEL FOR DIA1
0056 C VEL2 = VEL FOR DIA2
0057 C V0 = IMPACT VELOCITY (FT/SEC)
0058 C VR2 = RESIDUAL VELOCITY AFTER VESSEL WALL , FT/SEC
0059 C
0060 C
0061 C
0062 DIMENSION BHARD(3),C(3),DENS(3),FY(3),FSU(3),FTU(3),PDENS(3),
0063 . PFY(3),PFSU(3),PFTU(3),SHPV(3),SPACE(2),THICK(3),

```

```

0064      TARMAT(3)
0065      C
0066      INTEGER PROJMAT,TARMAT
0067      C
0068      LOGICAL PEN,INITIAL,SHATTER
0069      C
0070      REAL MASS
0071      C
0072      SAVE DIA1,DIA2,P2,VEL1,VEL2
0073      C
0074      C
0075      C SET INITIAL VALUES
0076      C
0077      I = 0
0078      N = 0
0079      PEN = .FALSE.
0080      PMINCR = 1.0E-04
0081      PI = 3.1415926536
0082      P1 = 1.0E-06
0083      C
0084      C CONVERT VARIABLES TO PEN4 FORMAT AND UNITS
0085      C
0086      C
0087      PROJMAT=2
0088      SPACE(1) = STAND/12.0
0089      SPACE(2) = 0.0
0090      TARMAT(1) = 1
0091      TARMAT(2) = 2
0092      TARMAT(3) = 0
0093      THETA = ANGR*180.0/PI
0094      THICK(1) = SHTHK/12.0
0095      THICK(2) = VWTHK/12.0
0096      THICK(3) = 0.0
0097      C
0098      C PEN4 DOES NOT SOLVE FOR THE CRITICAL PROJECTILE DIAMETER DIRECTLY
0099      C THEREFORE AN ITERATIVE APPROACH IS TAKEN. PEN4 ONLY RETURNS
0100      C A LOGICAL PARAMETER INDICATING IF PENETRATION HAS OCCURED.
0101      C
0102      C
0103      C FOR INITIAL CASE SET P2 > 0.0
0104      C
0105      IF (INITIAL) THEN
0106      P2 = 1.0E-04
0107      ENDIF
0108      C
0109      C IF SHATTER HAS OCCURED NO NEED TO CALL PEN4 , USE LINEAR APPROX.
0110      C
0111      C
0112      C DETERMINE INITIAL MASS THAT PENETRATES
0113      C
0114      100 MASS = P2 + N * PMINCR
0115      C
0116      NC = NC + 1
0117      C
0118      CALL NPEN4 ( VELE,MASS,THETA,THICK,SPACE,PEN,SHATTER,PROJMAT,
0119      1          DENS,FTU,C )
0120      C
0121      IF ( PEN ) THEN
0122      C
0123      C IF PENETRATION TRY WITH MASS BETWEEN P1 & P2
0124      C
0125      PEN = .FALSE.
0126      P2 = MASS

```

D180-30550-4

```

0127      MASS = ( P1 + P2 ) / 2.0
0128      C
0129      NC = NC + 1
0130      CALL NPEN4 ( VELE,MASS,THETA,THICK,SPACE,PEN,SHATTER,PROJMAT,
0131      1          DENS,FTU,C )
0132      C
0133      C
0134      200      IF ( PEN ) THEN
0135      C
0136      C          IF PENETRATION , SET P2 = MASS & CHECK RATIO
0137      C
0138      C          PEN = .FALSE.
0139      C          P2 = MASS
0140      C          RATIO = P2 /P1
0141      C          IF ( RATIO .LT. .995 .OR. RATIO .GT. 1.005 ) THEN
0142      C
0143      C          IF TRUE TRY AGAIN BETWEEN P1 & P2
0144      C
0145      C          MASS = ( P1 + P2 ) / 2.0
0146      C          NC = NC + 1
0147      C          CALL NPEN4 (VELE,MASS,THETA,THICK,SPACE,PEN,SHATTER,
0148      1          PROJMAT,DENS,FTU,C )
0149      C          GO TO 200
0150      C          ELSE
0151      C
0152      C          IF FALSE , MASS IS WITHIN RANGE SO CONTINUE
0153      C
0154      C          GO TO 300
0155      C          END IF
0156      C
0157      C          ELSE
0158      C
0159      C          IF FALSE , SET P1 = MASS AND CHECK RATIO
0160      C
0161      C          P1 = MASS
0162      C          RATIO = P2 / P1
0163      C          IF ( RATIO .LT. .995 .OR. RATIO .GT. 1.005 ) THEN
0164      C
0165      C          IF TRUE , TRY AGAIN BETWEEN P1 & P2
0166      C
0167      C          MASS = ( P1 + P2 ) / 2.0
0168      C          NC = NC + 1
0169      C          CALL NPEN4 (VELE,MASS,THETA,THICK,SPACE,PEN,SHATTER,
0170      1          PROJMAT,DENS,FTU,C)
0171      C          GO TO 200
0172      C          ELSE
0173      C
0174      C          IF FALSE , MASS IS WITHIN RANGE SO CONTINUE
0175      C
0176      C          GO TO 300
0177      C          END IF
0178      C          ENDIF
0179      C          ELSE
0180      C
0181      C          IF FALSE INCREASE INCREMENT & TRY AGAIN
0182      C
0183      C          P1 = MASS
0184      C          I = I + 1
0185      C          N = 2 ** I
0186      C          GO TO 100
0187      C          END IF
0188      C
0189      C          P2=MASS

```

D180-30550-4

```
0190      C
0191      C  CALCULATE DIAMETER
0192      C
0193      C
0194      C
0195      300  DEN = DENS(3)*1728.0
0196      C
0197          DIA = ((MASS*6.0)/(PI*DEN))**(1.0/3.0)
0198      C
0199      C  CONVERT TO IN.
0200      C
0201          DIA = DIA * 12.0
0202      C
0203      C
0204      C
0205      C  RESET HOLDERS
0206      C
0207          DIA2=DIA1
0208          DIA1=DIA
0209          VEL2=VEL1
0210          VEL1=VELE
0211      C
0212      C
0213          RETURN
0214      C
0215          END
```

```

0001      C                               D180-30550-4
0002      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0003      C
0004      SUBROUTINE NPEN4 (Vil,MProj,Thetal,Thick1,Space,Pennon,
0005      :                   Shater,PrMat1,Densel,YStrn1,SoundV)
0006      C
0007      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008      C
0009      C
0010      C
0011      C      PrMat Integer Projectile Material 1:Steel, 2:Aluminum ,3:Ice
0012
0013      CHARACTER Shape*3
0014      LOGICAL Pennon,Shater
0015      INTEGER Maxk(5)
0016      INTEGER PrMat,PrMat1,TarMat(10),TMatSp(10),Plate,Bin,NBin,I
0017      REAL RF(5),RC(5)
0018      REAL NF,J,MR,MProj,LastSp,LRM,FrMass(5)
0019      REAL Diam,Vi,Vr,ViLRM,VrLRM,Epsil,Gamma,Vil,Pi,Theta,SumSp
0020      REAL A,B,D,R,X,Y,ToverD,Rh,PlateM,FTheta,AllMas,Vc,DelJ,DelJ2
0021      REAL P,EffP,Pet,Pgrady,Thetal,AvgMas,Rp,F1,Vf
0022      REAL      Thick(10),Space(9),Thick1(10),
0023      :          PDense(3),PYStrn(3),PSondV(3),FrTuff(3),
0024      :          Dense(10),YStren(10),SoundV(10)
0025      REAL Densel(10),YStrn1(10)
0026      REAL ViX,MrMax,MProjX
0027      DOUBLE PRECISION Intact,HoArea
0028      DOUBLE PRECISION SumPr(5)
0029      DOUBLE PRECISION Nr,Pcr,Lambda,SigSq,Sigma,Pk,As,Ac
0030
0031      COMMON /Prob/Nr(5),Pcr(5),Lambda(5),SigSq(5),Sigma(5)
0032      COMMON /Crater/As,Ac(5),P(5)
0033      C
0034      DATA PDense/          Steel   Aluminum   Ice
0035      DATA PYStrn/          15.11 ,      5.39 ,      1.94/
0036      DATA PSondV/          8.35E+07 , 5485000.,      0.0/
0037      DATA FrTuff/          14960. ,    17569.,      0.0/
0038      DATA FrTuff/          36. ,      39.,      39./
0039      C      PrMat Projectile Material 1:Steel, 2:Aluminum ,3:Ice
0040      C      Calculation units
0041      C      Dense in Slugs/CubicFoot      Density
0042      C      YStren in Lbs/SquareFoot      Uniaxial Ultimate Stress
0043      C      SoundV in Feet/Second      Speed of Sound
0044      C      Theta in Radians      Impact Angle
0045      C      FrTuff in MegaPascals Meter^.5      Fracture Toughness
0046      C      Input units
0047      C      Densel input in Lbs/CubicInch
0048      C      YStrn1 input in PSI
0049      C      SoundV input in Feet/Sec
0050      C      Vil Feet/Sec      Impact Velocity
0051      C      MProj Pounds      Projectile Mass
0052      C      Thetal Degrees      Striking Angle
0053      C      Thick Feet      Target Plate Thickness
0054      C      Space Feet      Spacing Between Target Plates
0055      C      Outputs
0056      C      Pennon Logical      Penetration Flag
0057      C      Shater Logical      Fragment Shatter Flag
0058
0059      DATA Gamma/1.5708/
0060      DATA F1/4.0/
0061      DATA NBin/5/
0062      DATA TarMat/1,2,3,4,5,6,7,8,9,10/
0063      DATA TMatSp/2,2,2,2,2,2,2,2,2,2/
0064      DO 10 Plate=1,10

```

D180-30550-4

```

0064      Thick(Plate)=Thick1(Plate)
0065      Dense(Plate)=Dense1(Plate)/32.2*1728.
0066      YStren(Plate)=YStrn1(Plate)*144.
0067  10    CONTINUE
0068      Theta = Theta1/57.3
0069      PrMat = PrMat1
0070  C      END DO
0071      Vi=Vi1
0072      PI=3.141592653589793
0073      Shater =.FALSE.
0074  C      *****Calculate Radius of Projectile Sphere*****
0075      RP=(MProj*3./ (PDense(PrMat)*32.2*4.*PI))**(1./3.)
0076      Diam = RP * 2.
0077  C      *****
0078
0079      DO 2040 Plate=1,10
0080          IF (.NOT.Shater) THEN
0081  C          *** This option is for single penetrators*****
0082  C          **Compute Residual Velocity
0083
0084          Call ResVel
0085          :          (Vr,Vi,RP,PDense(PrMat),Dense(TarMat(Plate)),
0086          :          Thick(Plate),Theta,PrMat,TMatSp(Plate))
0087
0088  C          *****IF No Penetration Report Result and Exit*****
0089          IF (Vr.EQ.0.) THEN
0090              Pennon = .FALSE.
0091              RETURN
0092          ELSE IF (Thick(Plate+1).EQ.0.0.OR.Plate.EQ.10) THEN
0093              Pennon = .TRUE.
0094              RETURN
0095          ENDIF
0096  C          *****
0097  C          *****Mass Loss Regime Decision BAC IR&D*****
0098          ToverD = Thick(Plate)/Diam
0099          IF (PrMat.EQ.2.AND.TMatSp(Plate).EQ.2) THEN
0100              IF (ToverD.LT..1) THEN
0101                  Vf = 1116*ToverD**(-.55)
0102              ELSE
0103                  Vf = 4757*ToverD**.08
0104              END IF
0105          ELSE IF (PrMat.EQ.1.AND.TMatSp(Plate).EQ.2.) THEN
0106              IF (Shape.EQ.'CYL') THEN
0107                  Vf=5020* ToverD**.4
0108              ELSE IF (Shape.EQ.'CUB') THEN
0109                  Vf = AMAX1(1450*ToverD**(-.39),
0110                  :          4561*ToverD**42*(Diam*12*2.54)**(-.33))
0111              ELSE
0112                  Vf = AMAX1(1450*ToverD**(-.39),
0113                  :          4561*ToverD**.42*(Diam*12*2.54)**(-.33) )
0114
0115  C                  Vf = AMAX1(2362*ToverD**(-.35)*(Diam*12*2.54)**(-.32),
0116  C                  :          3937*ToverD**.23 *(Diam*12*2.54)**(-.25))
0117              END IF
0118          ELSE IF (PrMat.EQ.1.AND.TMatSp(Plate).EQ.1.) THEN
0119              Vf = 7021*ToverD**.39
0120          END IF
0121  C          *****
0122  C          RH=RP*(1.372E-4*Vi*(THICK(Plate)/(2.*RP))**(2./3.)+.9)*
0123  C          :          (1-EXP(-(1.48-Theta)/.166))
0124          RH = .5*Thick(Plate)*11.02*(1-EXP(-(1.48-Theta)/.166))*
0125          :          (1-EXP(-(PDense(PrMat)*Vi**2/YStren(TarMat(Plate))))**.415
0126          :          *(PDense(PrMat)/Dense(TarMat(Plate)))**(-.15)/ToverD/29.9))

```

D180-30550-4

```

0127      RH = AMAX1(RH,RP)
0128      IF (Vi.GT.Vf) THEN
0129          SumSp=Space(Plate)
0130          Shater=.TRUE.
0131      C      *** Compute plate spall *****
0132          PlateM=PI*RH**2/COS(Theta)*Thick(Plate)*
0133      :          Dense(TarMat(Plate))*32.2
0134      C      *****
0135      C      *****Hydrocode Predicted Mass Loss Due to Impact Angle*****
0136          FTHETA=-2.433E-4 - 1.643E-2*COS(THETA)+3.201*COS(THETA)**2-
0137      :          2.184*COS(THETA)**3
0138      C      *****
0139      C      *** COMPUTE FRAGMENT NUMBERS*****
0140          IF (PrMat.EQ.3) THEN
0141              MProj=PlateM
0142              RP=(PlateM*3./ (Dense(TarMat(Plate))*32.2*4.*PI))**(1./3.)
0143          END IF
0144
0145          CALL MasChr (Vi,MProj,ToverD,Theta,RP,PDense(PrMat),
0146      :          Dense(TarMat(Plate)),PSondV(PrMat),NBin,
0147      :          RF,NR,LRM,AvgMas,PrMat,FrTuff(PrMat))
0148
0149          IF (PrMat.EQ.3) THEN
0150              PrMat=TMatSp(Plate)
0151          ELSE
0152              AllMas=PlateM+MProj
0153              DO 2010 I=1,NBin-1
0154                  FrMass(I)=4./3.*PI*RF(I)**3*PDense(PrMat)*32.2
0155                  NR(I)=NR(I)*AllMas/MProj
0156      2010      CONTINUE
0157          END IF
0158          Nr(NBin) = 1.D0
0159          ViLRM=Vr
0160      C      *****Calculate Spray Angle*****
0161      C          Vc =11155*ToverD**(-.52)
0162      C          Epsil = 45*(1-EXP((- (Vi-Vf)/Vc)))/57.3
0163          Vc = 4889 * ToverD**(-.23)
0164          Epsil = 52*(1-EXP((- (Vi-Vf)/Vc)))/57.3
0165      C      *** Assurance of ThetaR+Epsil<90 and Spray Area finite*****
0166          ThetaR = AMIN1(Theta,1.41-Epsil)
0167      C      *** Calculate Spray Area*****
0168          DELJ=RP/2.0*(COS(EPSIL)-TAN(ThetaR)*SIN(EPSIL))
0169          DELJ2=RP*(1.0-TAN(ThetaR)*TAN(EPSIL))
0170          J=Space(Plate)*SIN(GAMMA)/SIN(ThetaR+GAMMA)
0171          X=(J*SIN(EPSIL)/2.+DELJ)*(1./SIN(ThetaR+GAMMA+EPSIL))-
0172      :          1./SIN(ThetaR+GAMMA-EPSIL))
0173          Y=X*SIN(ThetaR+GAMMA)
0174          R=(J-X*COS(ThetaR+GAMMA))*TAN(EPSIL)+DELJ2
0175          B=SQRT(R**2-Y**2)
0176          A=(J*SIN(EPSIL)/2.+DELJ)*(1./SIN(ThetaR+GAMMA-
0177      :          EPSIL)+1./SIN(ThetaR+GAMMA+EPSIL))
0178          AS=PI*A*B
0179      C      *****
0180      C      *** Allowance for increased penetration due to spalling of ***
0181      C      *** next plate *****
0182          EffP=1.7
0183      C      *****
0184      ELSE
0185          IF (PrMat.EQ.3) THEN
0186              MProj=3.1415926*Rh**2*Thick(Plate)
0187      :          *Dense(TarMat(Plate))*32.2
0188              PrMat=TMatSp(Plate)
0189          END IF

```

D180-30550-4

```

0190      Vi=Vr
0191      END IF
0192      C      *****
0193      ELSE
0194
0195      C      *****The rest of this subroutine contains the evaluation*
0196      C      ***** of shatter and fracture regime multiple cratering*
0197
0198      C      *** Allowance for increased penetration due to spalling of ***
0199      C      *** next plate *****
0200      Thick(Plate) = Thick(Plate)/EffP
0201      C      *****
0202
0203      C      *** COMPUTE PLATE CRATER Depth and Radius*****
0204      DO 2030 I=1,NBin
0205          D=2.*RF(I)
0206          PGrady=0.281*D*(PDense(PrMat)/Dense(TarMat(Plate)))**(1./3.)
0207          :      *(PDense(PrMat)*(Vr*COS(Theta))**2/
0208          :      (2.*YStren(TarMat(Plate))))**0.31
0209          IF (PrMat.EQ.1) THEN
0210              Pet = D*(Vr*COS(Theta)/12468.))**1.32
0211              P(I) = AMIN1(Pet,PGrady)
0212              IF (Vr.LT.3.9*3281) THEN
0213                  RC(I) = P(I)/2.*(Vr/3281./3.8)**(-1.32)
0214              ELSE
0215                  RC(I) = P(I)/2.*(Vr/3281./4.6)**.2
0216              END IF
0217          ELSE
0218              Pet = D*2.33E-5*(Vr*COS(Theta))**1.16
0219              P(I) = AMIN1(Pet,PGrady)
0220              RC(I) = P(I)/(1-EXP(-Vr/5578.))
0221          END IF
0222      C      *****
0223
0224      C      *** COMPUTE AVERAGE IMPACTS WITHIN CRATER*****
0225      AC(I)=PI*RC(I)**2/F1
0226      AC(I)=DMIN1(AC(I),.9999999999999999D0*AS)
0227      PCR(I)=DMIN1(1D0,AC(I)/AS)
0228      LAMBDA(I)=NR(I)*AC(I)/AS
0229      SigSq(I) = Lambda(I)*(1D0-PCR(I))
0230      Sigma(I) = SQRT(SigSq(I))
0231      C      *****
0232      2030      CONTINUE
0233      C      *** New Version eight section*****
0234      C      *** This subroutine finds the minimum number of each***
0235      C      *** size particle that must impact in one crater to ***
0236      C      *** penetrate the plate ,how many craters they are in,*
0237      C      *** and how many fragments are involved in shallower***
0238      C      *** craters.*****
0239      CALL PenK(Plate,Thick,NBin,Maxk)
0240      CALL Countr
0241      :      (NBin,P,Thick(Plate),Maxk,Intact,Nr,Ac,As)
0242      C      *** Number and Area of Holes and Residual Particles****
0243      HoArea=As*(1.D0-Intact)
0244      PlateM = HoArea*Thick(Plate)*Dense(Plate)*32.2
0245      AllMas=0.
0246      DO 2110 I=1,NBin-1
0247          AllMas=FrMass(I)*Nr(I)+AllMas
0248      2110      CONTINUE
0249      IF (AllMas.GT.0) THEN
0250          DO 2210 I=1,NBin-1
0251              NR(I)=NR(I)*(1.+PlateM/AllMas)
0252      2210      CONTINUE

```

```

0253      END IF
0254      C      *** Separate calculation for LRM***
0255      C      *** This option is for single penetrators*****
0256
0257      Call ResVel
0258      :      (Vr,ViLRM,RF(NBin),PDense(PrMat),Dense(TarMat(Plate)),
0259      :      Thick(Plate),Theta,PrMat,TMatSp(Plate))
0260
0261      C      ** Convert V to Km/S and MProj to grams
0262      ViX = Vi/3281.
0263      MProjX = LRM * 454.
0264
0265      C      ** Largest Residual Mass **
0266      ToverD=Thick(Plate)/2./Rf(NBin)
0267      CALL RFMax(MProjX,PDense(PrMat),ViX,RF(NBin),ToverD,MRMax,
0268      :      PSondV(PrMat),PrMat,Vc,Theta,FrTuff(PrMat))
0269      LRM=MrMax/454
0270      RF(NBin)=(LRM*3./(PDense(PrMat)*32.2*4.*PI))**(1./3.)
0271
0272      C      *****
0273      C      *** Test for Pennon and End of Run*****
0274      IF (HoArea.LT..000000069.AND.Vr.LE.0.00001) THEN
0275      C      IF (HoArea.LT.AC(1)) THEN
0276      Pennon = .FALSE.
0277      RETURN
0278      ELSE IF (Plate.EQ.10.OR.Thick(Plate+1).EQ.0.0) THEN
0279      Pennon = .TRUE.
0280      RETURN
0281      END IF
0282      ViLRM=Vr
0283      LastSp=SumSp
0284      SumSp=SumSp+Space(Plate)
0285      As=As*(SumSp/LastSp)**2
0286      C      *****
0287      C      *****END OF SHATTER EVALUATION*****
0288      END IF
0289      2040 CONTINUE
0290      RETURN
0291      END

```

```

0001
0002      SUBROUTINE RfMax(M,RhoP,V,RP,ToverD,MrMax,C,PrMat,Vc,Theta,Kic)
0003      C      M      Initial Impactor Mass      Grams
0004      C      MrMax  Mass of the Largest Residual Particle      Same as above
0005      C      RhoP   Impactor Density      Slugs/Ft^3
0006      C      Rp     Initial Impactor Radius (equivalent sphere) Ft
0007      C      V      Impactor Velocity      Km/Sec
0008      C      Shock  Velocity      Ft/Sec
0009      C      Kic    Fracture Toughness      MPa m^.5
0010      C      Theta  Approach angle      Radians
0011      C      PrMat  Projectile Material 1:Steel, 2:Aluminum ,3:Ice
0012      REAL Kic,k,MrMax,M,MrOMs,MrOMsC
0013      REAL MrMaxP,MrOMsP
0014      REAL MrMaxT,MrOMsT
0015      REAL RhoP,V,RP,ToverD,C,FTovrD,Vc
0016      INTEGER PrMat
0017      k=4.18E6
0018      IF (ToverD.LT..1) THEN
0019          FTovrD=1180.
0020      ELSE IF (ToverD.GE..1.AND.ToverD.LT..2) THEN
0021          FTovrD=697*ToverD**(-.23)
0022      ELSE IF (ToverD.GE.:2.AND.ToverD.LT..4) THEN
0023          FTovrD=244*ToverD**(-.881)
0024      ELSE IF (ToverD.GE..4.AND.ToverD.LT..8) THEN
0025          FTovrD=1500*ToverD**1.1
0026      ELSE
0027          FTovrD=1170.
0028      END IF
0029      IF (PrMat.EQ.1) THEN
0030          FTovrD=FTovrD*7.78
0031      END IF
0032      MrOMs = k*(Kic/(RhoP*32.2*C))**2/V**2/(RP*2*12*2.54)*FTovrD
0033      MrOMsT = MrOMs*COS(Theta)**(-2)
0034      MrOMsP = MrOMs*COS(Theta)
0035      Vc =(8.47E4*MrOMs*V**2)**(1/10.93)
0036      IF (V.LT.Vc) THEN
0037          MRMaxP=M*MrOMsP
0038      ELSE
0039          MrOMsC=1.18E-5*(Vc)**8.93
0040          MrMaxP=M*MrOMsC*(V/Vc)**(-5.5)
0041      END IF
0042      IF (V*COS(Theta).LT.Vc) THEN
0043          MRMaxT=M*MrOMsT
0044      ELSE
0045          MrOMsC=1.18E-5*(Vc)**8.93
0046          MrMaxT=M*MrOMsC*(V*COS(Theta)/Vc)**(-5.5)
0047      END IF
0048      MrMax = AMAX1(MrMaxP,MrMaxT)
0049      IF (PrMat.EQ.1) MrMax=MrMaxP
0050      MrMax = AMIN1(MrMax,M*.9999)
0051      RETURN
0052      END

```

D180-30550-4

```

0001
0002 SUBROUTINE PenK(Plate,Thick,NBin,Maxk)
0003 DOUBLE PRECISION Nr,Pcr,Lambda,SigSq,Sigma,Pk(5),SumPr(5),Pk2
0004 DOUBLE PRECISION As,Ac,TestAc
0005 INTEGER Bin,Plate,MaxK(5),I,BinsDo,NBin
0006 REAL Thick(10),P,Vc
0007 LOGICAL BINDON(5)
0008 INTEGER k(5),MinI
0009 DOUBLE PRECISION kProbs(5,0:15)
0010 REAL LnPk,LnkFac
0011 COMMON /Count/kProbs
0012 COMMON /Prob/Nr(5),Pcr(5),Lambda(5),SigSq(5),Sigma(5)
0013 COMMON /Crater/As,Ac(5),P(5)
0014
0015 C *** k(NBin) is the number of each particle size that must ****
0016 C *** impact in one crater to make a hole.*****
0017 C *** This loop finds the fraction of the plate that is not hit by *****
0018 C *** a fragment of size I (P(0)). BinDon are initialized***
0019 TestAc=1D0-1D-3*Ac(1)/As
0020 BinsDo=0
0021 DO 10 I=1,NBin
0022     k(I)=0D0
0023     CALL Prs(I,Pk(I),k,NBin)
0024     kProbs(I,0)=Pk(I)
0025     SumPr(I)=Pk(I)
0026 C *** If there is not room for one more crater on the plate then *****
0027 C *** stop using this particle size *****
0028     IF (SumPr(I).GT.TestAc.OR.
0029 :      (K(I)*P(I)).GT.Thick(Plate)) THEN
0030         BinDon(I)=.TRUE.
0031         BinsDo=BinsDo+1
0032         kProbs(I,1)=1D0-kProbs(I,0)
0033     ELSE
0034         BinDon(I)=.FALSE.
0035     END IF
0036 C *****
0037 10 CONTINUE
0038 C END DO
0039 C *****
0040 C *** This subroutine finds which fragment size has the least SumPr *****
0041 C *** SumPr is the fraction of the plate that has craters of depth <= k**
0042 C *****
0043 C *** This loop sums up the area of the plate that is not penetrated,****
0044 C *** while keeping the area covered by craters of depth <=k for each ***
0045 C *** size approximately equal.*****
0046 20 IF (BinsDo.GE.NBin-1) GOTO 30
0047     CALL MiniI(MinI,BinDon,NBin,SumPr)
0048     k(MinI) = k(MinI)+1
0049 C *** This subroutine Calculates the fraction of the plate that is ****
0050 C *** covered by craters from exactly k particles of size MinI *****
0051     CALL PrS(MinI,Pk(MinI),k,NBin)
0052 C *****
0053 C *** Add up the fraction of the spray area accounted for, and the ****
0054 C *** fraction of the particles used so far*****
0055     SumPr(MinI)=SumPr(MinI)+Pk(MinI)
0056     kProbs(MinI,k(MinI))=Pk(MinI)
0057 C *****
0058 C *** If there is not room for one more crater on the plate then *****
0059 C *** stop using this particle size*****
0060     IF (SumPr(MinI).GT.TestAc.OR.
0061 :      (K(MinI)*P(MinI)).GT.Thick(Plate)) THEN
0062         BinDon(MinI)=.TRUE.
0063         BinsDo=BinsDo+1

```

D180-30550-4

```
0064      Maxk (MinI) = K (MinI)
0065      END IF
0066 C      *****
0067      GOTO 20
0068 30    CONTINUE
0069      K (NBin) = 1
0070      kProbs (NBin, 1) = 1D0 - kProbs (NBin, 0)
0071      Maxk (NBin) = 1
0072 C      *****
0073      RETURN
0074      END
```

```

0001
0002      SUBROUTINE MiniI(MinI,BinDon,NBin,SumPr)
0003      LOGICAL BinDon(5)
0004      INTEGER BinsDo
0005      DOUBLE PRECISION SumPr,MinSum
0006      DIMENSION SumPr(5)
0007      MinSum=1D0
0008      DO 10 I=1,NBin-1
0009          IF (.NOT.BinDon(I).AND.SumPr(I).LT.MinSum) THEN
0010              MinI=I
0011              MinSum=SumPr(I)
0012          END IF
0013      10 CONTINUE
0014      C      END DO
0015      RETURN
0016      END
0001
0002      SUBROUTINE PrS(I,Pk,k,NBin)
0003      DOUBLE PRECISION LastTe(5),PkMin,Fpi
0004      DOUBLE PRECISION Nr,Pcr,Lambda,SigSq,Sigma,Pk,SumPr
0005      DOUBLE PRECISION As,Ac
0006      DIMENSION k(5)
0007      DIMENSION PkMin(5),kMin(5)
0008      COMMON /Prob/Nr(5),Pcr(5),Lambda(5),SigSq(5),Sigma(5)
0009      Fpi=1D0/SQRT(2D0*3.1415926D0)
0010      IF (Nr(I).GE.50.AND.Lambda(I).LT.5) THEN
0011      C      *** This section calculates the poisson approximation to ***
0012      C      *** the binomial distribution*****
0013          IF (K(I).EQ.0) THEN
0014      C      *** Initialize P(k) for first value*****
0015              Pk=DEXP(-Lambda(I))
0016      C      *****
0017          ELSE
0018      C      *** Calculate P(k) from P(k-1)*****
0019              Pk=LastTe(I)*Lambda(I)/K(I)
0020      C      *****
0021          END IF
0022          LastTe(I)=Pk
0023      C      *****
0024      ELSE IF (Pcr(I).GT..1.AND.Pcr(I).LT..9.AND.
0025      :      Nr(I).GT.9./(Pcr(I)*(1-Pcr(I)))) THEN
0026      C      *** This section calculates the normal approximation to ****
0027      C      *** the binomial distribution*****
0028          Pk=Fpi/Sigma(I)/DEXP(1/(2.D0*SigSq(I)*(k(I)-Lambda(I))**2))
0029      C      *****
0030      ELSE
0031      C      ***This section calculates the binomial distribution*****
0032          IF (k(I).EQ.0) THEN
0033      C      *** Find least non-zero P(k) and corresponding k*****
0034              CALL SetBin(I,PkMin,kMin,NBin)
0035              LastTe(I)=PkMin(I)
0036              IF (kMin(I).EQ.0) Pk=PkMin(I)
0037      C      *****
0038          ELSE IF (k(I).GT.kMin(I)) THEN
0039      C      *** calculate the non-zero P(k)s*****
0040              Pk = LastTe(I)*Pcr(I)/(1D0-Pcr(I))*(Nr(I)-k(I)+1D0)/k(I)
0041              LastTe(I)=Pk
0042      C      *****
0043          ELSE IF (k(I).EQ.KMin(I)) THEN
0044              Pk = PkMin(I)
0045          ELSE
0046              Pk = 0D0
0047          END IF

```

```
0048      C      *****
0049      END IF
0050      RETURN
0051      END
```

D180-30550-4

```

0001
0002      SUBROUTINE SetBin(I,PkMin,kMin,NBin)
0003      C      *** The Probability of the mean case occuring is computed *****
0004      C      *** in subroutine Binomial. P(kMin) is calculated from *****
0005      C      *** this P(Lambda). The magnitude of P(kMin) is arbitrarily ***
0006      C      *** chosen to be about 1*10^-7.*****
0007
0008      COMMON /Prob/Nr(5),Pcr(5),Lambda(5),SigSq(5),Sigma(5)
0009
0010      DOUBLE PRECISION Nr,Pcr,Lambda,SigSq,Sigma,Pk,SumPr,N1,PkMin,P1
0011      DIMENSION PkMin(5),kMin(5)
0012      k=Lambda(I)
0013      N1=Nr(I)
0014      P1=Pcr(I)
0015      CALL Binomi(k,N1,Pk,P1)
0016      1  IF (.NOT.(Pk.GT.1D-13.AND.k.GE.1)) GOTO 10
0017          Pk = Pk*(1D0-Pcr(I))/Pcr(I)*k/(Nr(I)-k-1D0)
0018          k = k-1
0019      GOTO 1
0020      10  CONTINUE
0021      C  END DO
0022          kMin(I)=k
0023          PkMin(I)=Pk
0024      RETURN
0025      END

```

D180-30550-4

```

0001
0002      SUBROUTINE Binomi (k1,Nr,Pk,Pcr)
0003      DOUBLE PRECISION Nr
0004      DOUBLE PRECISION Pk,Pcr,Lambda,LastPk,Qcr
0005      DOUBLE PRECISION Top,Lower1,Lower2,k,Expon
0006      Lambda=k1
0007      k=k1
0008      C      *****Binomial Distribution, Calculates P(k)*****
0009      Lower1 = 0d0
0010      Lower2 = 0d0
0011      Top = Nr-k
0012      Pk = 1d0
0013      Qcr = 1D0-Pcr
0014      LastPk = 0D0
0015      IF (Qcr.EQ.1.D0) THEN
0016          IF (k.EQ.0.D0) THEN
0017              Pk=1.D0
0018          ELSE
0019              Pk=0.D0
0020          ENDIF
0021      ELSE
0022          3040  IF (Lower1+Lower2+Top.GE.(2*Nr).OR.Pk.EQ.LastPk) GOTO 3030
0023              LastPk = Pk
0024          3060  IF (Top.GE.Nr.OR.Pk.GE.1E23) GOTO 3050
0025              Top = Top + 1D0
0026              Pk = Pk * Top
0027              GOTO 3060
0028          3050  CONTINUE
0029          3080  IF (Lower1.GE.k.OR.Pk.LE.1E-20) GOTO 3070
0030              Lower1 = Lower1 + 1d0
0031              Pk = Pk /Lower1*Pcr
0032              GOTO 3080
0033          3070  CONTINUE
0034              IF (Lower1.GE.k.AND.Lower2.LT.Nr-k.AND.Pk.GT.1D-27) THEN
0035                  Expon=DMIN1((-28.D0-DLOG10(Pk))/DLOG10(Qcr),(Nr-k-Lower2))
0036                  IF (Expon.GT.0) Pk=Pk*Qcr**Expon
0037                  Lower2 = Lower2+Expon
0038              ENDIF
0039              GOTO 3040
0040          3030  CONTINUE
0041      END IF
0042      C      *****
0043      RETURN
0044      END

```

D180-30550-4

```

0001
0002      SUBROUTINE MasChr (V,M,ToverD,Theta,RP,RhoP,RhoT,C,NBin,
0003      :                  MenRad,Nr,MrMax,MAvg,PrMat,FrTuff)
0004      DOUBLE PRECISION NR(5)
0005      REAL MProj,M,MRMax,MAvg,MenRad(5),MPlate
0006      REAL MasLim,NrmF,NrmS,BinMen(5),BinMas(0:5)
0007      INTEGER Bin,PrMat
0008      PI = 3.14159
0009
0010      C      This subroutine divides the residual mass into bins of equal mass
0011      C      The number of fragments in each bin are also noted
0012
0013      C      ** Convert V to Km/S and MProj to grams
0014      Vi = V/3281.
0015      MProj = M * 454.
0016      MPlate = M * 454.
0017
0018      C      ** Largest Residual Mass **
0019
0020      IF (PrMat.EQ.3) THEN
0021          CT = 17569
0022          RhoT = 5.39
0023          CALL RFMax(MPlate,RhoT,Vi,RP,ToverD,MRMax,
0024      :              CT,PrMat,Vc,Theta,FrTuff)
0025      ELSE
0026          CALL RFMax(MProj,RhoP,Vi,RP,ToverD,MRMax,
0027      :              C,PrMat,Vc,Theta,FrTuff)
0028      END IF
0029
0030      C      ** Average Residual Mass **
0031      CALL AvgRes (ToverD,Theta,MPlate,Vi,Alfa,MAvg)
0032      IF (Vi.GT.Vc) MAVg=MAvg*(Vi/Vc)**(-5.5)
0033
0034      C      ** Parameters and Normalization Constants for Weibull Distribution **
0035      CALL ShCons (ToverD,Theta,MProj,Vi,MAvg,
0036      :              bS,sS,bF,sF,NrmS,NrmF,MRMax)
0037
0038      C      ** Size Shatter Begins **
0039      CALL Shhold (MProj,Vi,TD,Theta,MAvg,bF,sF,bS,sS,
0040      :              NrmF,NrmS,FrgLim,MasLim,MRMax)
0041
0042      C      ** Upper Bound and Mean Fragment weight for each Bin **
0043      CALL BinLim(NBin,MAvg,MRMax,BinMas,BinMen,MProj,
0044      :              sF,bF,sS,bS,NrmS,NrmF,MasLim)
0045
0046      C      ** Loop Determining Number of Fragments in each Bin in Shatter Regime *
0047      DO 10 Bin = 1,NBin-1
0048          Nr(Bin)=(BinMas(Bin)-BinMas(Bin-1))/BinMen(Bin)
0049      10 CONTINUE
0050      C      END DO
0051      Nr(NBin) = 1
0052      C      ** Convert Bin Masses into lbs from grams then to Radius in Feet**
0053      DO 20 Bin = 1,NBin
0054          BinMen(Bin)=BinMen(Bin)/454.
0055          MenRad(Bin) = (BinMen(Bin)*3./(RhoP*32.2*4.*PI))**(1./3.)
0056      20 CONTINUE
0057      C      END DO
0058      MrMax=MrMax/454.
0059      MPlate = MPlate/454.
0060      MProj = MProj/454.
0061      RETURN
0062      END

```

D180-30550-4

```

0001
0002      SUBROUTINE MasDis (M,MAvg,b,s,NrmC,MasSum)
0003      C
0004      C      THIS PROGRAM COMPUTES SHATTER Mass distributions
0005      C      STEEL CUBES ON ALUMINUM PLATE DATA.
0006      C      V = IMPACT VELOCITY      (FEET/SECOND)
0007      C      TD = T OVER D
0008      C      M = PROJECTILE MASS      (GRAINS)
0009      C      THETA = ANGLE OF OBLIQUITY
0010      C      ALPHA=ANGLE OF IMPACT
0011      C
0012      REAL MAvg,M,NrmC,MasSum
0013      W = M/MAvg
0014      C      **Mass in Bin**
0015      determ=b+s*LOG(W)
0016      IF (Determ.LT.-80) THEN
0017          MasSum=0
0018      ELSE IF (Determ.LT'.4) THEN
0019          MasSum = NrmC*(1-EXP(-EXP(Determ)))
0020      ELSE
0021          MasSum=NrmC
0022      END IF
0023      RETURN
0024      END

```

D180-30550-4

```

0001      C      SSSSSSSSSSSSSS  SALVO's  CHANGES  SSSSSSSSSSSSSS
0002
0003      SUBROUTINE ShCons(TD,Theta,M,Vi,MAvg,
0004      :                      bS,sS,bF,sF,NrmCS,NrmCF,LRM)
0005      REAL M,NrmCS,NrmCF,MT2,MAvg,IntegS,IntegF,LRM
0006      sS=1.8-.04*TD/COS(THETA)-.042*M
0007      &      +.34*(COS(2.*ALPHA)**2.)-(1-EXP(-.64*Vi))
0008      bS=-2.3-1.1*TD/COS(THETA)+.0675*M
0009      &      -.27*Vi+1.4*(COS(2.*ALPHA)**2.)
0010      sF=1.38-.510*TD/COS(THETA)+.036*M
0011      &      +3.31*(COS(2.*ALPHA)**2.)-(1-EXP(-.390*Vi))
0012      bF=-1.17+.313*TD/COS(THETA)+.0675*M
0013      &      +.508*Vi-1.41*(COS(2.*ALPHA)**2.)
0014
0015      C      ** Total mass in shatter regime **
0016      MT2 =(-.957+EXP(-.0013*M))*(Vi**(.38*M+2.5))
0017      MT2 = AMAX1(MT2,1E-20)
0018      MT2 = AMIN1(MT2,M)
0019      C      **Normalization to Largest Residual Mass**
0020      C      IntegS = 1-EXP(-EXP(bS)*(M/MAvg)**sS)
0021      C      IntegF = 1-EXP(-EXP(bF)*(M/MAvg)**sF)
0022      C      NrmCS = MT2/IntegS
0023      C      NrmCF = M /IntegF
0024      detrml=bS+sS*ALOG(LRM/MAvg)
0025      IF (Detrml.LT.-15) THEN
0026      NrmCS=1e-30
0027      ELSE IF (Detrml.LT.4) THEN
0028      IntegS = 1-EXP(-EXP(Detrml))
0029      NrmCS = MT2/IntegS
0030      ELSE
0031      NrmCS = MT2
0032      END IF
0033      detrml2=bF+sF*LOG(LRM/MAvg)
0034      IF (Detrml2.LT.-15) THEN
0035      NrmCS=1e-30
0036      ELSE IF (Detrml2.LT.4) THEN
0037      IntegF = 1-EXP(-EXP(Detrml2))
0038      NrmCF = M/IntegF
0039      ELSE
0040      NrmCF = M
0041      END IF
0042      if(sF)1,2,2
0043      1      sF=0.
0044      2      if(sS)3,4,4
0045      3      sS=0.
0046      4      RETURN
0047      END

```

D180-30550-4

```

0001
0002      SUBROUTINE Shhold (MProj,Vi,TD,Theta,MAvg,bF,sF,bS,sS,
0003      :                   NrmF,NrmS,FrgLim,MasLim,LRM)
0004      REAL LowLim,Mean,Mfs,Mss,MAvg,MasLim,MProj,NrmF,NrmS,M,LRM
0005      HiLim = MProj
0006      LowLim = 0
0007      DO 10 I=1,20
0008          Mean = (HiLim+LowLim)/2
0009          CALL MasDis (Mean,MAvg,bF,sF,NrmF,Mfs)
0010          CALL MasDis (Mean,MAvg,bS,sS,NrmS,Mss)
0011          IF (Mfs.GT.Mss) THEN
0012              HiLim = Mean
0013          ELSE
0014              LowLim = Mean
0015          END IF
0016      10 CONTINUE
0017      C      END DO
0018          FrgLim = Mean
0019          MasLim = AMIN1 (Mss,MProj-LRM)
0020      RETURN
0021      END

```

D180-30550-4

```

0001
0002 SUBROUTINE BinLim (NBin,MAvg,LRM,BinMas,BinMen,MProj,
0003 : sF,bF,sS,bS,NrmS,NrmF,MasLim)
0004 REAL MAvg,LRM,MProj,MasLim,NrmS,NrmF,BinInt,MasMen,MFrac(0:4)
0005 DIMENSION BinMen(5),BinMas(0:5),MasMen(5)
0006 INTEGER Bin,ShtBin
0007 DATA MFrac/0.,.25,.5,.75,1./
0008 ShtBin=0
0009 DO 5 Bin=1,NBin-1
0010     BinMas(Bin) = (MProj-LRM)*MFrac(Bin)
0011     MasMen(Bin) = (MProj-LRM)*(MFrac(Bin)+MFrac(Bin-1))/2.
0012     IF (BinMas(Bin).LE.MasLim) ShtBin=Bin
0013 5 CONTINUE
0014 C END DO
0015 C ** Bin Limit and Mean for Shatter **
0016 DO 10 Bin = 1,ShtBin
0017     BinMen(Bin) = MAvg*(-EXP(-bS)*DLOG(1D0-MasMen(Bin)/NrmS))**(1/sS)
0018 10 CONTINUE
0019 C END DO
0020 C ** Bin Limit and Mean for First Bin in Fracture **
0021 Bin = ShtBin+1
0022 IF (Bin.LE.NBin-1) THEN
0023     IF (MasLim.GE.MasMen(Bin)) THEN
0024 C         ** Bin Mean in Case it is Less Than the Fracture Threshold **
0025         BinMen(Bin)=MAvg*(-EXP(-bS)*DLOG(1D0-MasMen(Bin)/NrmS))**(1/sS)
0026     ELSE
0027         BinMen(Bin)=MAvg*(-EXP(-bF)*DLOG(1D0-MasMen(Bin)/NrmF))**(1/sF)
0028     END IF
0029 END IF
0030 C ** Bin Limit and Mean for Fracture **
0031 DO 20 Bin = ShtBin+2,NBin-1
0032     BinMen(Bin) = MAvg*(-EXP(-bF)*DLOG(1D0-MasMen(Bin)/NrmF))**(1/sF)
0033 20 CONTINUE
0034 C END DO
0035 C ** Bin Mean for largest bin is LRM (Largest Residual Mass) **
0036 BinMen(NBin) = LRM
0037 RETURN
0038 END

```

```

0001
0002 SUBROUTINE AvgRes (ToverD,Theta,M,V,Alfa,MAvg)
0003 REAL MAvg,M
0004 C ** Average Fragment Mass **
0005 MAvg = .0109-.00879*ToverD/COS(Theta)+.000506*M-.00428*V+
0006 : .0110*COS(2*Alfa)**2
0007 MAvg = AMax1(MAvg,0.005)
0008 RETURN
0009 END

```

D180-30550-4

```

0001
0002      SUBROUTINE Countr(NBin,P,Thick,Maxk,Intact,Nr,Ac,As)
0003      INTEGER Bin,NBin,SmBin
0004      INTEGER Digit(5),HoMin(5),I,Maxk(5)
0005      REAL Thick,HoDept,P(5)
0006      DOUBLE PRECISION Probs(5,0:15)
0007      DOUBLE PRECISION Intact,ITArea,ITArS1
0008      DOUBLE PRECISION Nr(5),Ac(5),As
0009      DOUBLE PRECISION ArHol1,ArHol2,ArHol3
0010      DOUBLE PRECISION ArHol4,ArHol5
0011      DOUBLE PRECISION PrTemp,DigTp1,DigTp2,DigTp3
0012      DOUBLE PRECISION          DigTp4,DigTp5
0013      LOGICAL UnInc,OBinld
0014      COMMON /Count/Probs
0015      DO 10 Bin = 1,NBin
0016          HoMin(Bin) = JMIN0(INT(Thick/P(Bin)+1),Maxk(Bin))
0017          Digit(Bin)=0
0018      10  CONTINUE
0019      C      END DO
0020          ArHol1=0.D0
0021          ArHol2=0.D0
0022          ArHol3=0.D0
0023          ArHol4=0.D0
0024          ArHol5=0.D0
0025          Intact=0.D0
0026          Bin = 0
0027          HoDept = 0
0028          OBinld = .FALSE.
0029      C      DO WHILE (Bin.LE.NBin)
0030      2    IF (.NOT.(Bin.LE.NBin)) GOTO 20
0031          IF (HoDept.GT.Thick) THEN
0032              OBinld = .FALSE.
0033              SmBin=1
0034      C          DO WHILE (Digit(SmBin).EQ.0)
0035      21    IF (.NOT.(Digit(SmBin).EQ.0)) GOTO 210
0036              SmBin=SmBin+1
0037              GOTO 21
0038      210  CONTINUE
0039      C      END DO
0040          Digit(SmBin)=0
0041          HoDept=0.
0042          Bin = SmBin+1
0043          UnInc = .TRUE.
0044      C      DO WHILE (Bin.LE.NBin.AND.UnInc)
0045      22    IF (.NOT.(Bin.LE.NBin.AND.UnInc)) GOTO 220
0046          IF (Digit(Bin).LT.HoMin(Bin)) THEN
0047              Digit(Bin) = Digit(Bin)+1
0048              UnInc = .FALSE.
0049          ELSE
0050              Digit(Bin)=0
0051              Bin = Bin + 1
0052          END IF
0053          GOTO 22
0054      220  CONTINUE
0055      C      END DO
0056          DO 230 I=SmBin+1,NBin
0057              HoDept=HoDept+Digit(I)*P(I)
0058      230  CONTINUE
0059      C      END DO
0060      ELSE
0061          IF (OBinld) THEN
0062              DigTp1=DigTp1+1
0063              ITArea = Probs(1,DigTp1)*ITArS1

```

D180-30550-4

```

0064
0065      ArHol1 = ArHol1+ITArea*DigTp1
0066      ArHol2 = ArHol2+ITArea*DigTp2
0067      ArHol3 = ArHol3+ITArea*DigTp3
0068      ArHol4 = ArHol4+ITArea*DigTp4
0069      ArHol5 = ArHol5+ITArea*DigTp5
0070
0071      ELSE
0072          IArea=1.D0
0073
0074          DigTp2=Digit(2)
0075          PrTemp = Probs(2,DigTp2)
0076          IArea = PrTemp*ITArea
0077
0078          DigTp3=Digit(3)
0079          PrTemp = Probs(3,DigTp3)
0080          IArea = PrTemp *ITArea
0081
0082          DigTp4=Digit(4)
0083          PrTemp = Probs(4,DigTp4)
0084          IArea = PrTemp *ITArea
0085
0086          DigTp5=Digit(5)
0087          PrTemp = Probs(5,DigTp5)
0088          IArea = PrTemp *ITArea
0089
0090          IArS1=ITArea
0091
0092          DigTp1=Digit(1)
0093          PrTemp = Probs(1,DigTp1)
0094          IArea = PrTemp *ITArea
0095
0096          ArHol1 = ArHol1+ITArea*DigTp1
0097          ArHol2 = ArHol2+ITArea*DigTp2
0098          ArHol3 = ArHol3+ITArea*DigTp3
0099          ArHol4 = ArHol4+ITArea*DigTp4
0100          ArHol5 = ArHol5+ITArea*DigTp5
0101
0102      END IF
0103      Intact=Intact+ITArea
0104      Bin = 1
0105      OBinld = .TRUE.
0106      UnInc = .TRUE.
0107      C      DO WHILE (Bin.LE.NBin.AND.UnInc)
0108          24      IF (.NOT.(Bin.LE.NBin.AND.UnInc)) GOTO 240
0109              IF (Digit(Bin).LT.HoMin(Bin)) THEN
0110                  Digit(Bin) = Digit(Bin)+1
0111                  HoDept = HoDept+P(Bin)
0112                  UnInc = .FALSE.
0113              ELSE
0114                  OBinld = .FALSE.
0115                  HoDept = HoDept-Digit(Bin)*P(Bin)
0116                  Digit(Bin)=0
0117                  Bin = Bin + 1
0118              END IF
0119              GOTO 24
0120          240      CONTINUE
0121      C      END DO
0122      END IF
0123      GOTO 2
0124      20      CONTINUE
0125      C      END DO
0126      Nr(1)=DMAX1(0D0,Nr(1)-ArHol1*As/Ac(1))

```

D180-30550-4

```
0127      Nr(2)=DMAX1(OD0,Nr(2)-ArHol2*As/Ac(2))
0128      Nr(3)=DMAX1(OD0,Nr(3)-ArHol3*As/Ac(3))
0129      Nr(4)=DMAX1(OD0,Nr(4)-ArHol4*As/Ac(4))
0130      Nr(5)=DMAX1(OD0,Nr(5)-ArHol5*As/Ac(5))
0131      RETURN
0132      END
```

D180-30550-4

```

0001
0002      SUBROUTINE ResVel
0003      :      (Vr,V0,RP,RhoP,RhoT,Thick,Theta,PrMat,TarMat)
0004      C      This is the JTCG Residual Velocity Formula
0005      INTEGER PrMat,TarMat
0006      PresAr = 3.14159*RP**2
0007      V0cm = V0/.03281
0008      Weight = 4./3.*3.14159*RP**3*RhoP
0009      V50 = Ballim(RP,RhoP,RhoT,Thick,Theta,PrMat,TarMat)
0010      Q4 = RhoT*Thick*PresAr/(Weight*COS(Theta))
0011      VrSq = AMAX1(0.,V0cm**2-V50**2)
0012      Vr = SQRT(VrSq) / (1.+Q4)
0013      Vr = Vr*.03281
0014      RETURN
0015      END

0001
0002      FUNCTION Ballim
0003      :      (RP,RhoP,RhoT,Thick,Theta,PrMat,TarMat)
0004      INTEGER PrMat,TarMat
0005      IF (PrMat.EQ.3) THEN
0006          Ballim = 2.45*3281*(Thick/12./2.54)**(-.15)*
0007      :      (Thick/RP/2.)*(RhoT/RhoP)**.64*(1./COS(Theta))**1.01
0008      ELSE
0009      C      This is the JTCG V50 formula
0010          PresAr = 3.14159*RP**2
0011          Weight = 4./3.*3.14159*RP**3*RhoP*32.2
0012          W0 = .0143
0013          IF (PrMat.EQ.1) THEN
0014              IF (TarMat.EQ.2) THEN
0015                  Cbf = 41300.
0016                  Bf = .941
0017                  H = 1.098
0018                  F = -.038
0019              ELSE
0020                  Cbf = 80600.
0021                  Bf = .963
0022                  H = 1.286
0023                  F = -.057
0024              END IF
0025          ELSE
0026              Cbf = 92800
0027              Bf = .972
0028              H = 1.01
0029              F = -.105
0030          END IF
0031          Q8 = RhoP*32.2*Thick*PresAr/Weight
0032          Q11 = RhoP*32.2*Thick*PresAr/W0
0033          Ballim = Cbf*Q8**Bf/COS(Theta)**H*Q11**F
0034      END IF
0035      RETURN
0036      END

```

```

0001      C                                D180-30550-4
0002      C
0003      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0004      C
0005      C          SUBROUTINE CRITDIA
0006      C
0007      CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
0008      C
0009      C
0010      C Critdia determines the diameter of the sphere that just penetrates
0011      C the given wall configuration at the given impact angle and velocity.
0012      C The subroutine performs a linear interpolation using the appropriate
0013      C values from the Response array to estimate the critical diameter.
0014      C
0015      C note: for variables contained in the common block refer to the main
0016      C listing for definition
0017      C
0018      C
0019      C Variable list
0020      C
0021      C     beta = impact angle , measured from the normal , deg
0022      C     b1 = the impact angle nearest to the actual impact angle
0023      C           in the Response array , but less than the actual , deg
0024      C     ib1 = location of b1 in the Response array
0025      C     ib2 = " " " b2 " " " "
0026      C     iv1 = " " " v1 " " " "
0027      C     iv2 = " " " v2 " " " "
0028      C     r1 = intermediate variable
0029      C     r2 = " "
0030      C     r11 = value in Response array at location iv1,ib1,pid
0031      C     r12 = " " " " " " " iv1,ib2,pid
0032      C     r21 = " " " " " " " iv2,ib1,pid
0033      C     r22 = " " " " " " " iv2,ib2,pid
0034      C     v1 = impact velocity nearest the actual impact velocity in the
0035      C           Response array , but still less than the actual
0036      C
0037      C
0038      C
0039      C
0040      C     INTEGER*2 IB1,IB2,IV1,IV2,PID
0041      C
0042      C     INCLUDE 'COMMON3.BLK'
0043      C
0044      C
0045      C     PARAMETER (PI=3.1415926536)
0046      C
0047      C Determine the location of the nearest velocity to the actual velocity
0048      C in the Response array, but still less than the actual
0049      C
0050      C     IV1=VR/VINC
0051      C
0052      C Check that the location is inside the array
0053      C
0054      C     IF ( IV1.LT.1 .OR. IV1.GT.NV ) THEN
0055      C         WRITE ( 6,10 )VR
0056      C         10 FORMAT ( /1X,'THE RELATIVE VELOCITY (VR) IS OUTSIDE OF THE',
0057      C             1 ' RESPONSE ARRAY BOUNDS VR (KM/SEC) = ',E12.5)
0058      C         STOP
0059      C     END IF
0060      C
0061      C Set the location of the velocity just greater than the actual velocity
0062      C
0063      C     IV2=IV1+1
0064      C
0065      C

```

D180-30550-4

```

0086 C Check that the location is inside the array
0087 C
0088 IF ( IV2.GT.NV ) IV2=IV1-1
0089 C
0090 C Calculate the value of the velocity at location iv1
0091 C
0092 V1=IV1*VINC
0093 C
0094 C Determine the impact angle in deg
0095 C
0096 BETA=ACOS(CBETA)*180.0/PI
0097 C
0098 C Determine the location of the nearest impact angle to the actual
0099 C impact angle in the Response array, but still less than the actual
0100 C
0101 IB1=BETA/BINC+1
0102 C
0103 C Check that the location is inside the array
0104 C
0105 IF ( IB1.LT.1 .OR. IB1.GT.NB ) THEN
0106 WRITE ( 6,20 )BETA
0107 20 FORMAT ( /1X,'IMPACT ANGLE (BETA) IS OUTSIDE THE BOUNDS OF',
0108 1 ' THE RESPONSE ARRAY BETA (DEG) = ',E12.5)
0109 STOP
0110 END IF
0111 C
0112 C Set the location of the impact angle in the Response array that is
0113 C just greater than the actual
0114 C
0115 IB2=IB1+1
0116 C
0117 C Check that the location is inside the array
0118 C
0119 IF ( IB2.GT.NB ) IB2=IB1-1
0120 C
0121 C Calculate the value of the impact angle at location ib1 in the Response
0122 C array
0123 C
0124 B1=(IB1-1)*BINC
0125 C
0126 C Determine the property id
0127 C
0128 PID=ID(2,NEL)
0129 C
0130 C Get the four values that surround the actual value in the Response
0131 C array
0132 C
0133 R11=RTABLE(IV1,IB1,PID)
0134 R12=RTABLE(IV1,IB2,PID)
0135 R21=RTABLE(IV2,IB1,PID)
0136 R22=RTABLE(IV2,IB2,PID)
0137 C
0138 C Using linear interpolation, estimate the critical diameter
0139 C
0140 R1=(R12-R11)*((BETA-B1)/BINC)+R11
0141 R2=(R22-R21)*((BETA-B1)/BINC)+R21
0142 C
0143 DIAM=(R2-R1)*((VR-V1)/VINC)+R1
0144 C
0145 C Finished , return
0146 C
0147 RETURN
0148 C

```

0149

END

D180-30550-4

```

0001 C
0002 C D180-30550-4
0003 C
0004 C
0005 C SUBROUTINE FLUX
0006 C
0007 C
0008 C
0009 C
0010 C Flux calculates the meteoroid or debris flux for the given critical
0011 C diameter based on analysis type.
0012 C
0013 C
0014 C note: for variables contained in the common block refer to the main
0015 C listing for definition
0016 C
0017 C Variable List
0018 C
0019 C ddiam = diam in double precision , cm
0020 C ge = gravity focusing factor
0021 C intercept = intercept of the flux equation
0022 C mass = critical meteoroid mass, g
0023 C mden = meteoroid density, g/cc
0024 C re = earth's radius (including 100km atmosphere), km
0025 C slope = slope of the flux equation
0026 C
0027 C
0028 C INCLUDE 'COMMON3.BLK'
0051 C
0052 C REAL*8 DDIAM,GE,INTERCEPT,LD,MASS,MDEN,PI,RE,SLOPE
0053 C
0054 C PARAMETER (PI=3.141592653589793238D0)
0055 C
0056 C Set mden
0057 C
0058 C
0059 C MDEN=0.50D0
0060 C
0061 C Calculate the focusing factor, equation
0062 C is from JSC-30000
0063 C
0064 C RE=6478.0D0
0065 C GE=0.568D0+0.432D0*(RE/(RE+ALT))
0066 C
0067 C Convert diam to double precision
0068 C
0069 C DDIAM=DIAM
0070 C
0071 C Calculate the flux
0072 C
0073 C IF ( ITYPE.EQ.1 ) THEN
0074 C
0075 C For debris use JSC-20001, use stated equations for diameters
0076 C less than 1 cm , for those greater use third order fit of the
0077 C curve for region up to 5 cm .
0078 C
0079 C The log of the flux varies linearly between 400 and 500 km according
0080 C to D Kessler of JSC.
0081 C
0082 C LD=DLOG10(DDIAM)
0083 C IF ( DIAM.LT.5.0 ) THEN
0084 C IF ( DIAM.LT.1.0 ) THEN
0085 C SLOPE=-0.0010D0*ALT-2.0200D0

```

D180-30550-4

```

0086             ELSE
0087             SLOPE=-0.0022D0*ALT-0.1400D0
0088             END IF
0089             INTERCEPT=+0.0036D0*ALT-7.26D0
0090             FLX=10.0D0** (SLOPE*LD+INTERCEPT)
0091             ELSE
0092             WRITE ( 6,100 )
0093 100           FORMAT ( /1X,'DIAMETER IS GREATER THAN 5 CM LIMIT')
0094             STOP
0095             END IF
0096 C
0097 C   Correct Flux for differance in Boeing and Nasa definetion
0098 C
0099             FLX=FLX*4.0D0
0100 C
0101             ELSE
0102 C
0103 C   For meteoroids use JSC-3000,  E-06g < mass < 1g
0104 C
0105             MASS=PI*(DDIAM**3)/6.0D0*MDEN
0106             FLX=10.0D0**(-14.37D0-1.213D0*DLOG10(MASS))
0107 C
0108 C   Account for earth shielding and gravity focusing , also convert to
0109 C   number of impacts per sq-m per year
0110 C
0111             FLX=FLX*GE*3.15576D07
0112 C
0113             END IF
0114 C
0115             RETURN
0116 C
0117             END

```

COMMON3.BLK

```

C
C Common Block for Contour Ver 2.0 6/2/87
C
C ielm = max number of elements
C ith = max number of threats
C
C   PARAMETER (IELM=9000, ITH=400)
C
C   INTEGER*2 IT, ITYPE, NB, NC, NEL, NELM, NT, NV, EXPOSED(ITH),
1     POINT(IELM, ITH)
C
C   INTEGER*4 NR, ID(2, IELM), RANGE(2)
C
C   REAL*4 BINQ, CBETA, DIAM, VINQ, AREA(IELM), GEOMETRY(IELM, ITH),
1     RTABLE(70, 90, 10), THREAT(4, ITH)
C
C   REAL*8 ALT, ETIME, FLX
C
C   COMMON ALT, BINQ, CBETA, DIAM, ETIME, FLX, IT, ITYPE, NB, NC, NEL, NELM, NR,
1     NT, NV, VR, VINQ, AREA, EXPOSED, GEOMETRY, ID, POINT, RANGE, RTABLE,
2     THREAT
C

```